YD JA 000232

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９９年　１月１４日

出願番号
Application Number: 平成１１年特許願第００７３８４号

出願人
Applicant (s): インターナショナル・ビジネス・マシーンズ・コーポレイション

CERTIFIED COPY OF
PRIORITY DOCUMENT

１９９９年　４月３０日

特許庁長官
Commissioner,
Patent Office 伊佐山建志

出証番号　出証特平１１－３０２７５２０

【書類名】　特許願

【整理番号】　JA998232

【あて先】　特許庁長官　殿

【国際特許分類】　H03M 13/00

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　田村　哲也

【特許出願人】

【識別番号】　390009531

【住所又は居所】　アメリカ合衆国１０５０４、ニューヨーク州アーモンク（番地なし）

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレイション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【連絡先】　０４６２－７３－３３１８、３３２５、３４５５

【選任した代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【手数料の表示】

【予納台帳番号】　024154

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

出証特平１１－３０２７５２０

【物件名】　　要約書　1

【包括委任状番号】　9304391

【包括委任状番号】　9304392

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　標数２のガロア体上で定義される超楕円曲線のヤコビ多様体の群演算を実施する装置及び方法

【特許請求の範囲】

【請求項１】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）及びD_2=g.c.d.（$a_2(x)$，$y-b_2(x)$）に対し群演算を実施する装置であって、

$a_1(x)$、$a_2(x)$、$b_1(x)$及び$b_2(x)$を格納する手段と、

GCD（$a_1(x)$，$a_2(x)$）＝１（GCDは最大公約多項式）である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$又は$s_2(x)$を用いて、$q(x)=\{s_1(x)(b_1(x)+b_2(x))\}$ｍｏｄ $a_2(x)$又は$q(x)=\{s_2(x)(b_1(x)+b_2(x))\}$ｍｏｄ $a_1(x)$を計算する手段と、

を有することを特徴とする装置。

【請求項２】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）及びD_2=g.c.d.（$a_2(x)$，$y-b_2(x)$）に対し、D_1+D_2と線形同値なリデュースト・ディバイザD'=g.c.d.（a'(x)，y－b'(x)）のa'（ｘ）及びb'（ｘ）を計算する装置であって、

GCD（$a_1(x)$，$a_2(x)$）＝１（GCDは最大公約多項式）である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$を用いて、$q(x)=s_1(x)(b_1(x)+b_2(x))$ｍｏｄ $a_2(x)$を計算する手段と、

モニック化されたα（ｘ）＝Ｑ（$q^2(x)a_1(x)$，$a_2(x)$）＋Ｑ（ｆ（ｘ），$a_1(x)a_2(x)$）（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段と、

β（ｘ）＝（$q(x)a_1(x)+b_1(x)+1$）ｍｏｄ α（ｘ）を計算する手段と、

a'（ｘ）＝Ｑ（ｆ（ｘ）＋$\beta^2(x)$，α(x)）を計算する手段と、

b'（ｘ）＝（$\beta(x)+1$）ｍｏｄ a'(x)を計算する手段と、

を有する装置。

【請求項３】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）に対し群演算を実施する装置であって、

$a_1(x)$及び$b_1(x)$を格納する手段と、

$q(x)=Q(b_1^2(x)+f(x) \bmod a_1^2(x), a_1(x))$（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段、

を有することを特徴とする装置。

【請求項４】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）に対し、D_1+D_1と線形同値なリデュースト・ディバイザD'=g.c.d.（a'(x)，$y-b'(x)$）のａ'（ｘ）及びｂ'（ｘ）を計算する装置であって、

$q(x)=Q(b_1^2(x)+f(x) \bmod a_1^2(x), a_1(x))$（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段と、

モニック化された$\alpha(x)=q^2(x)+Q(f(x), a_1^2(x))$を計算する手段と、

$\beta(x)=(b_1^2(x)+f(x) \bmod a_1^2(x)+1) \bmod \alpha(x)$を計算する手段と、

$a'(x)=Q(f(x)+\beta^2(x), \alpha(x))$を計算する手段と、

$b'(x)=(\beta(x)+1) \bmod a'(x)$を計算する手段と、

を有する装置。

【請求項５】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）及びD_2=g.c.d.（$a_2(x)$，$y-b_2(x)$）に対し、D_1+D_2と線形同値なリデュースト・ディバイザD'=g.c.d.（a'(x)，$y-b'(x)$）のａ'（ｘ）及びｂ'（ｘ）を計算する方法であって、

GCD（$a_1(x)$，$a_2(x)$）＝１（GCDは最大公約多項式）である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$を用いて、$q(x)=\{s_1(x)(b_1(x)+b_2(x))\} \bmod a_2(x)$を計算し且つ記憶装置に格納するステップと、

モニック化された$\alpha(x)=Q(q^2(x)a_1(x), a_2(x))+Q(f(x), a$

$_1(x)\,a_2(x)$）（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算し且つ記憶装置に格納するステップと、

β（ｘ）＝（ｑ(x)a_1(x)＋b_1(x)＋１）ｍｏｄ α（ｘ）を計算し且つ記憶装置に格納するステップと、

ａ’（ｘ）＝Ｑ（ｆ（ｘ）＋β^2(x)，α(x)）を計算し且つ記憶装置に格納するステップと、

ｂ’（ｘ）＝（β(x)＋１）ｍｏｄ ａ’(x)を計算し且つ記憶装置に格納するステップと、

を含む方法。

【請求項６】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1＝g.c.d.（a_1(x)，ｙ－b_1(x)）に対し、D_1+D_1と線形同値なリデュースト・ディバイザD’＝g.c.d.（ａ’(x)，ｙ－ｂ’(x)）のａ’（ｘ）及びｂ’（ｘ）を計算する方法であって、

ｑ（ｘ）＝Ｑ（b_1^2(x)＋ｆ（ｘ） mod a_1^2(x)，a_1）（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算し且つ記憶装置に格納するステップと、

モニック化されたα（ｘ）＝q^2（ｘ）＋Ｑ（ｆ（ｘ），a_1^2(x)）を計算し且つ記憶装置に格納するステップと、

β（ｘ）＝（b_1^2(x)＋ｆ（ｘ） mod a_1^2(x)＋１）ｍｏｄ α（ｘ）を計算し且つ記憶装置に格納するステップと、

ａ’（ｘ）＝Ｑ（ｆ（ｘ）＋β^2(x)，α(x)）を計算し且つ記憶装置に格納するステップと、

ｂ’（ｘ）＝（β(x)＋１）ｍｏｄ ａ’(x)を計算し且つ記憶装置に格納するステップと、

を含む方法。

【請求項７】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1＝g.c.d.（a_1(x)，ｙ－b_1(x)）及びD_2＝g.c.d.（a_2(x)，ｙ－b_2(x)）に対し群演算を実施する方法であって、

$a_1(x)$、$a_2(x)$、$b_1(x)$及び$b_2(x)$を格納するステップと、

GCD（$a_1(x)$，$a_2(x)$）＝１である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$又は$s_2(x)$を用いて、$q(x)=s_1(x)(b_1(x)+b_2(x)) \bmod a_2(x)$又は$q(x)=\{s_2(x)(b_1(x)+b_2(x))\} \bmod a_1(x)$を計算し且つ記憶装置に格納するステップと、

を含む、方法。

【請求項８】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）に対し群演算を実施する方法であって、

$a_1(x)$及び$b_1(x)$を格納するステップと、

$q(x)=Q(b_1^2(x)+f(x) \bmod a_1^2(x), a_1)$（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算し且つ記憶装置に格納するステップ、

を含む、方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）及びD_2=g.c.d.（$a_2(x)$，$y-b_2(x)$）に対し群演算を実施する装置及び方法に関する。

【０００２】

【従来の技術】

本願は、ＧＦ（2^n）上で定義される超楕円曲線のヤコビ多様体の群演算をハードウェアで実行するのに適したアルゴリズムを開示する。以下、本発明を理解する上で必要な前提知識について説明する。

【０００３】

［１］超楕円曲線及び因子

体をＫ、その代数的閉体を$\overline{K}$（Ｋの上にバーが付いたもの）とする。Ｋ上、種数ｇの超楕円曲線Ｃは式：$y^2+h(x)y=f(x)$で定義される。ここで$h(x)$は、次数ｇ以下の多項式であり、$f(x)$は２ｇ＋１以下のモニック多項式である。

多項式ｆ及びｈの係数はＫの元である。曲線Ｃは特異点を持たないものとする。また有理点Ｐ＝（ｘ，ｙ）が与えられた時、Ｐの共役点をP^{-}＝（ｘ，－ｙ－ｈ(x)）と定義する（P^{-}はＰの上にバーが付いたもの）。もしＰが無限遠点P_{∞}ならば、$P_{\infty}=P_{\infty}^{-}$（$P_{\infty}^{-}$は$P_{\infty}$の上にバーが付いたもの）とする。以後、本願では体Ｋ＝ＧＦ（2^{n}）、ｈ(x)＝１の場合について考える。

【０００４】

曲線Ｃ上の有限個のＫ$^{-}$有理点$P_1,\ldots P_r$についての整数係数の形式的な和$m_1P_1+m_2P_2+\ldots+m_rP_r$を考えて、これをＣの因子(divisor)と呼ぶ。Ｃの因子は一般的には

【数１】

$$D=\sum_{P_i\in C} m_i P_i$$

で与えられる。因子Ｄの次数は、ｄｅｇＤ＝Σm_iとする。Ｃの因子

【数２】

$$D_1 = \sum_{P_i \in C} m_i P_i$$

【数３】

$$D_2 = \sum_{P_i \in C} n_i P_i$$

に対して、

【数４】

$$D_1 + D_2 = \sum_{P_i \in C} (m_i + n_i) P_i$$

と定義することにより、Ｃの因子全体の集合Ｄ（Ｃ）は加群となり、これを因子群と呼ぶ。次数が０の因子全体は因子群の部分群となり、これをD^0（Ｃ）と記

す。曲線Ｃの０でない有理関数ｈの零点及び極は高々有限個であることより、ｈの零点及び極を用いてｈの因子div(h)を

【数５】

$$div(h)=\sum_{P_i\in C} ord_{P_i}(h)P_i=\sum m_iP_i-\sum n_iQ_i$$

で定義する。ここでP_iは有理関数ｈの零点であり、m_iはその重複度、Q_iは有理関数ｈの極であり、n_iは極の重複度、ord_{Pi}（ｈ）は有理関数ｈの点P_iにおける位数である。ある有理関数の因子となるような因子を主因子（principal divisor）という。主因子全体の集合を主因子群と呼び、D^1（Ｃ）と表す。

【０００５】

一般に、有理関数の零点の個数と極の個数は重複度（位数）を込めて考えると等しいのでD^1（Ｃ）$\subset D^0$（Ｃ）である。２つの因子D_1（数１），D_2（数２）$\in D^0$（Ｃ）が与えられた時、２つの因子のg.c.d.（D_1，D_2）を、$\Sigma\min(m_i, n_i)P_i-(\Sigma\min(m_i, n_i)P_\infty$で定義する。）。また式より明らかにg.c.d.（$D_1$，$D_2$）$\subset D^0$（Ｃ）である。

【０００６】

［２］ヤコビ多様体の定義

ヤコビ多様体は、Ｃの次数が０の因子全体のなす群D^0（Ｃ）の主因子群D^1（Ｃ）に関する剰余群で定義される（山本芳彦，“数論入門2” 岩波書店（1996）等を参照のこと）。これをＪ（Ｃ）と表す。もし、D_1，$D_2\in D^0$（Ｃ）で$D_1-D_2\in D^1$（Ｃ）であるならば、D_1，D_2は線形同値$D_1\sim D_2$であるという。$\forall D\in D^0$（Ｃ）は、次の条件を満たす因子D_1

【数６】

$$D_1=\sum_{P_i\in C} m_iP_i-\left(\sum_{P_i\in C} m_i\right)P_\infty$$

（$m_i \geqq 0$）に変形することができる。

（１）$D_1 \sim D$

（２）$P_i = (x_i, y_i)$と共役点$\overline{P_i}$（P_iの上にバーが付いたもの）は同時に現れない。

（３）$P_i = \overline{P_i}$である時は、$m_i = 1$

このように変形された因子をセミリデュースト・ディバイザ（semi-reduced divisor）と呼ぶ。さらにsemi-reduced divisorに対し、

【数７】

$$\sum_{P_i \in C} m_i \leqq g$$

の条件を与えることにより、ヤコビ多様体の元は一意に表現される。このように変形された因子をリデュースト・ディバイザ（reduced divisor）と呼ぶ。

【０００７】

任意のsemi-reduced divisor Dは、D＝g.c.d.（a（x），y－b（x））によって一意に表すことができる。ここでa（x）$=\Pi_i (x - x_i)^{mi}$、b（x）はb（x_i）＝y_i、deg b＜deg aを満たす唯一の多項式である。Dがreduce

d divisorである必要十分条件は、deg a≦gである。以降、D.G.Cantor, "Computing in the Jacobian of a Hyperelliptic Curve," Math. of Comp, 48, No.177, pp.95-101, (1987) にならって、g.c.d.（a（x），y－b（x））をｄｉｖ（a，b）と表す。また、これ以後因子Dは２つの多項式a及びbと同一視する。

【０００８】

Ｊ（Ｃ；ＧＦ（2^n））上の離散対数問題とは、D_1，D_2∈Ｊ（Ｃ；ＧＦ（2^n））に対して、$D_1=mD_2$となる整数mを求める問題である。

【０００９】

［３］ヤコビ多様体の安全条件

安全な超楕円曲線暗号を構成するために，ヤコビ多様体Ｊ（Ｃ；ＧＦ（2^n））が満たさなければならない条件は、酒井康行，石塚裕一，櫻井幸一，“安全な超楕円曲線暗号の構成とその実装，” SCIS'98-10.1.B, Jan. 1998等によると、次のとおりである。

Ｃ１　＃Ｊ（Ｃ；ＧＦ（2^n））が大きな素数で割切れる。

Ｃ２　Ｊ（Ｃ；ＧＦ（2^n））の最大素因数が（2^n）$^k-1$（k＜（log2^2）^{2}を割り切らない。

Ｃ３　２ｇ＋１＜log2^n

【００１０】

［４］ヤコビ多様体の群演算のアルゴリズム

ヤコビ多様体における加算は、D_1，D_2∈Ｊ（Ｃ；ＧＦ（2^n））に対し、次にD_1+D_2と線形同値なreduced divisor D'を求めることである。Cantorの先の論文及び N.Koblitz, "Hyperelliptic curve cryptosystems," Jounal of Cryptology, 1, pp.139-150, (1989)によれば、加算のアルゴリズムは２つの手続きから成る。この手続き１では、入力D_1＝ｄｉｖ（a_1，b_1）及びD_2＝ｄｉｖ（a_2，b_2）に対し、D_1+D_2～D（D＝ｄｉｖ（a，b））となるようなsemi-reduced divisor Ｄを求める。手続き２ではこのＤを入力としてD～D'（D'＝ｄｉｖ（a'，b'），deg b'＜deg a'，deg a'≦g）となるような reduced divisor D'を求める。超楕円曲線を$y^2+h(x)y=f(x)$とすると、これらの

手続きは次のとおりになる。

【００１１】

手続き１

Input　a_1, b_1　$D_1 = div(a_1, b_1)$

　　a_2, b_2　$D_2 = div(a_2, b_2)$

Output　a, b

（１）$a_1(x)$, $a_2(x)$, $b_1(x)+b_2(x)+h(x)$の最大公約多項式（GCDと記する）$d=d(x)$であって、$d=s_1a_1+s_2a_2+s_3(b_1+b_2+h)$を満たす$s_1(x)$, $s_2(x)$, $s_3(x)$を求める。

（２）$a(x)$, $b(x)$を次式に基づいて計算する。

$a=a_1a_2/d^2$

$b=(s_1a_1b_2+s_2a_2b_1+s_3(b_1b_2+f))/d \mod a$

手続き２

Input　a, b

Output　a', b'　　D～D'

（１）次式に基づいて、$a'(x)$及び$b'(x)$を計算する。

$a'=(f-hb-b^2)/a$

$b'=(-h-b) \mod a'$

（２）if (deg a'＞g) then

　　a＝a'

　　b＝b'

　　goto (１)

　else end

特に２倍算では手続き１は次のように簡単化できる。

手続き１

$a = a_1^2$

$b = (b_1^2 + f) \bmod a$

goto 手続き２（１）

【００１２】

【発明が解決しようとする課題】

上で述べたようなアルゴリズムにてそのまま計算を実施すると、次数の高い多項式の演算が必要となり、計算量が多くなるという欠点がある。

【００１３】

よって、本発明の目的は、ヤコビ多様体における群演算を少ない演算量にて実現することとする。

【００１４】

また、ヤコビ多様体における群演算を少ないハードウエア量にて実施可能とすることも、本発明の目的である。

【００１５】

【課題を解決するための手段】

上でも引用した論文に記載されているようにKoblitzは、種数が１より大きい超楕円曲線のヤコビ多様体上での離散対数問題を用いた暗号を提案した。しかし、種数が２の曲線はFreyによって安全でないことが示された（G. Frey, H.G. Ruck, "A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves," Math. of Comp, 62,No.206, pp.865-874, (1994) を参照のこと）。種数が３以上の曲線では、安全と思われる曲線がいくつか見つかっている（酒井康行，石塚裕一，櫻井幸一，“安全な超楕円曲線暗号の構成とその実装，” SCIS'98-10.1.B, Jan. 1998、北村出，“小さい標数の有限体上でヤコビ多様体が almost prime となる超楕円曲線，” SCIS'98-7.1.A, Jan. 1998、S. Arita, "Public Key Cryptosystems with Cab curve(1)," IEICE ISEC97-54 pp. 13-23 (1997)等を参照のこと）。

一般にＧＦ（2^n）における演算は（１）加算・乗算が比較的小規模なハードウェアで高速に実行できる、（２）２乗演算が簡単にできる、（３）逆元演算が伊東－辻井の提案した方法によって高速に実行できる（T. Itoh, S.Tsujii, " A Fast Algorithm for Computing Multiplicative Inverse in GF(2m) Using Normal Bases," Inform. and Comput. , vol.83, No. 1, pp.171-177, (1989)）などの理由でハードウェア実装に適する。さらに使用する定義体が楕円曲線暗号と比べて小さくて済むこと、上で述べた、Cantorのアルゴリズムにおいて行われる最大公約多項式を求める計算が複数個の乗算器を並列実行させることによって効率的に行うことができることから、超楕円曲線暗号は楕円曲線暗号よりハードウェア実装にむいている。よって、本発明では、Cantorのアルゴリズムを改良して、計算量及びハードウエア量を減らす。

【００１６】

そのため以下のような特徴がある。すなわち、ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$, $y-b_1(x)$）及びD_2=g.c.d.（$a_2(x)$, $y-b_2(x)$）（g.c.d.は段落番号０００５に定義されている）に対し群演算を実施する装置は、$a_1(x)$、$a_2(x)$、$b_1(x)$及び$b_2(x)$を格納する手段と、GCD（$a_1(x)$, $a_2(x)$）＝１（GCDは最大公約多項式）である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$を用いて、$q(x)=\{s_1(x)(b_1(x)+b_2(x))\} \bmod a_2(x)$を計算する手段と有する。このように新たな関数ｑ（ｘ）を設けることにより、全体の計算量が減少し、且つハードウエア量も少なくて済む。実施例では、超楕円曲線が$y^2+y=x^7$である例が詳しく述べられているが、このような曲線でない場合であっても、このｑ（ｘ）は有効に用いることができる。なお、群演算では交換則が成立するので、a_1, b_1及びs_1と、a_2, b_2及びs_2とを交換して得られるｑ（ｘ）等を用いても同じ解を得ることができる。これ以降煩雑を防ぐために片方のみを用いて説明することもあるが、交換しても同じ意味を有する。$q(x)=\{s_1(x)(b_1(x)+b_2(x))\} \bmod a_2(x)$は、$q(x)=\{s_2(x)(b_1(x)+b_2(x))\} \bmod a_1(x)$とすることができる。

【００１７】

なお、$D_1=D_2$の場合には、$a_1(x)$及び$b_1(x)$を格納する手段と、ｑ（ｘ）＝Ｑ（$b_1^2(x)+f(x) \bmod a_1^2(x)$, a_1）（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段を設ける。このように別個のｑ（ｘ）を定義する。

【００１８】

超楕円曲線が$y^2+y=f(x)$である場合に、ＧＦ（2^n）上で定義される超楕円曲線のヤコビ多様体の因子D_1＝g.c.d.（$a_1(x)$, $y-b_1(x)$）及びD_2＝g.c.d.（$a_2(x)$, $y-b_2(x)$）に対し、D_1+D_2と線形同値なリデュースト・ディバイザD'＝g.c.d.（a'(x), $y-b'(x)$）のａ’（ｘ）及びｂ’（ｘ）を計算する装置は、GCD（$a_1(x)$, $a_2(x)$）＝１（GCDは最大公約多項式）である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$を用いて、ｑ（ｘ）＝$s_1(x)$（$b_1(x)+b_2(x)$）ｍｏｄ $a_2(x)$を計算する手段と、モニック化されたα（ｘ）＝Ｑ（$q^2(x)a_1(x)$, $a_2(x)$）＋Ｑ（ｆ（ｘ），$a_1(x)a_2(x)$）（又は、α（ｘ）＝Ｑ（$q^2(x)a_2(x)$, $a_1(x)$）＋Ｑ（ｆ（ｘ），$a_1(x)a_2(x)$））（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段と、β（ｘ）＝（$q(x)a_1(x)+b_1(x)+1$）ｍｏｄ α（ｘ）（又は、β（ｘ）＝（$q(x)a_2(x)+b_2(x)+1$）ｍｏｄ α（ｘ））を計算する手段と、ａ’（ｘ）＝Ｑ（ｆ（ｘ）＋$\beta^2(x)$, $\alpha(x)$）を計算する手段と、ｂ’（ｘ）＝（$\beta(x)+1$）ｍｏｄ a'(x)を計算する手段とを有する。

【００１９】

一方、$D_1=D_2$である場合には、ｑ（ｘ）＝Ｑ（$b_1^2(x)+f(x) \bmod a_1^2(x)$, a_1）（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段と、モニック化されたα（ｘ）＝q^2（ｘ）＋Ｑ（ｆ（ｘ），$a_1^2(x)$）を計算する手段と、β（ｘ）＝（$b_1^2(x)+f(x) \bmod a_1^2(x)+1$）ｍｏｄ α（ｘ）を計算する手段と、ａ’（ｘ）＝Ｑ（ｆ（ｘ）＋$\beta^2(x)$, $\alpha(x)$）を計算する手段と、ｂ’（ｘ）＝（$\beta(x)+1$）ｍｏｄ a'(x)を計算する手段とを有する。

【００２０】

なお、以上はハードウエア化を前提とした構成であるが、これらをコンピュータ・プログラム等で実行するように変形することも可能である。その際、プログラムは、フロッピー・ディスクやＣＤ－ＲＯＭ等の記憶媒体及び他の記憶装置に

格納される。

【００２１】

【発明の実施の形態】

まず、本発明の基本的なアルゴリズムを説明する。

【００２２】

なお、加算を行う時にはa_1とa_2の最大公約多項式を求めなければならない。しかし、定義体が大きく且つa_1とa_2が任意に選ばれた座標から与えられる時には、多くの場合GCD（a_1，a_2）＝１となる。それゆえa_1とa_2が素でない場合の処理は性能に大きな影響を与えないので、本願ではGCD（a_1，a_2）≠１の場合を以後扱わない。また、その最大公約多項式GCD（a_1，a_2）＝１は多項式s_1，s_2によって、$s_1(x)a_1(x)+s_2(x)a_2(x)=1$のように表されるものとする。なお、素でない場合の計算においても、後で説明する手続き１の簡単化に用いる補題１及び２を一般化したものと、手続き２の簡単化に用いられる関数Ｑ（ｕ，ｖ）を使用することができる。

【００２３】

又、上で説明した従来技術の手続き１（１）の演算、すなわち、多項式s_1，s_2と最大公約多項式を求めるのには通常ユークリッド法が用いられる。ユークリッド法は、リード・ソロモン（Reed-Solomon）符号などの復号で、誤り位置多項式・誤り数値多項式を求める時に使用され、多くの実装が行われている。例えば、特開平７－２０２７１８号公報又は特開昭６２－１２２３３２号公報を参照のこと。よって、本発明では、手続き１（１）の２つの多項式の最大公約多項式ｄとｄ＝$s_1a_1+s_2a_2$を満たすs_1（又はs_2）のみを求める処理は、計算できたものとして扱う。また、例えば以下で述べる実装例（図１）を用いて計算することもできる。実際の動作の詳細を説明するのは省略するが、$s_1(x)$はＹregに出力されるものとする。また最大公約多項式ｄが１となるよう$s_1(x)$は正規化されているものとする。

【００２４】

a_1とa_2が素である場合について、ヤコビ多様体の群演算のアルゴリズム（従来技術）を手続き１（２）以降について変形すると次のとおりとなる。

変形１（通常の加算の場合）

Input　a_1, a_2, b_1, b_2

Output　a', b'

$a_3(x)=a_1a_2$

$b_3(x)=(s_1a_1b_2+s_2a_2b_1) \mod a_3$

$a_4(x)=(f+b_3+b_3^2)/a_3$

$a_4(x)$をモニック化（最高次の項の係数を１にする）

$b_4(x)=(b_3+1) \mod a_4(x)$

while (deg $a_4(x)>g$) {

$a'=a_5(x)=(f+b_4+b_4(x)^2)/a_4(x)$

$b'=b_5(x)=(b_4+1) \mod a_5(x)$

$a_4(x)=a'$

$b_4(x)=b'$

} end

このアルゴリズムに対しｆ(x)＝x^7を用いる。群演算のアルゴリズムの手続き２（１）ではａ及びｂの次数は２下がり、それ以後１づつ次数が下がることから、while loopの中を１回実行することにより、次数が３以下のａ'が求まることが分かる。b_3の計算で剰余演算される多項式は７次式、$a_4(x)$の計算で被除多項式は１０次式となり，多くの計算が必要となる．これを減らすために新しい多項式 q（x）＝$s_1(b_1+b_2) \mod a_2$を導入する。

【００２５】

補題１

変形１の最初の$a_4(x)$は、q(x)を用いて$a_4(x)=Q(q^2a_1, a_2)$で与えられる。ここでＱ（ｕ，ｖ）はｕ／ｖの商を与える関数である。

（証明）

まず、$b_3(x)=q a_1+b_1$であることを示す。上で述べた仮定より、$s_1 a_1+s_2 a_2=1$。これを用いてb_3の計算を$\deg a_1 a_2 > \deg b_1$に注意して書き直すと、

$$
\begin{aligned}
b_3(x) &= (s_1 a_1 b_2+s_2 a_2 b_1) \bmod a_3 \\
&= (s_1 a_1 b_2+(1+s_1 a_1) b_1) \bmod a_1 a_2 \\
&= (s_1 a_1 (b_1+b_2)) \bmod a_1 a_2+b_1 \\
&= \{(s_1 (b_1+b_2)) \bmod a_2\} a_1+b_1 \\
&= q a_1+b_1 \quad \cdots\cdots\cdots\cdots (\#)
\end{aligned}
$$

次にa_4の割り算は割り切れることと、$\deg b_3 < \deg a_3$より、$Q(b_3, a_3)=0$であることから、

$$
\begin{aligned}
a_4 &= Q(f+b_3+b_3^2, a_3) \\
&= Q(f, a_3)+Q(b_3^2, a_3)
\end{aligned}
$$

である。第２項は（＃）を代入し、$\deg b_1^2 < \deg a_3$より$Q(b_1^2, a_3)=0$であることに注意すると、

$$Q(b_3^2, a_3)=Q(q^2 a_1^2+b_1^2, a_3)=Q(q^2 a_1^2, a_3)$$

これより、$a_4(x)=Q(q^2 a_1, a_2)+Q(f, a_3)$　Q.E.D.

【００２６】

入力多項式を、

$$a_1(x)=x^3+c_2 x^2+c_1 x+c_0$$
$$a_2(x)=x^3+e_2 x^2+e_1 x+e_0$$
$$b_1(x)=d_2 x^2+d_1 x+d_0$$
$$b_2(x)=f_2 x^2+f_1 x+f_0$$

と定義し、$f(x)=x^7$を用いると、a_4の第２項は、$Q(x^7, a_3)=x+c_2+e_2$となる。これより、$q(x)$を用いて変形１を書き直すと、以下のような本発明のアルゴリズムとなる。

【００２７】

本発明のアルゴリズム（加算）

Input　a_1, a_2, b_1, b_2

Output　a', b'

$q(x)=s_1(b_1+b_2) \mod a_2$

$a_4(x)=Q(q^2a_1, a_2)+x+c_2+e_2$

a_4をモニック化（最高次数の項の係数を１にする）

$b_4(x)=(q a_1+b_1+1) \mod a_4$

if (deg a_4 ＞3) then

$a'=a_5(x)=Q(x^7+b_4^2, a_4)$

$b'=b_5(x)=(b_4+1) \mod a_5$

else　$a'=a_4$, $b'=b_4$

end

【００２８】

なお$a_5(x)$の計算で、deg b_4 ＜deg a_4であることよりQ（b_4, a_4）＝０であることを用いている。このアルゴリズムでは、次数の高い$a_3(x)$が消えて、その剰余演算及び除算を行う必要がなくなっている。また$a_4(x)$を計算するのに必要な乗算は、Qの中のq^2a_1の次数が７次であることと標数が２であることより、９回でよい。さらに$a_5(x)$の計算において必要のない$b_4(x)$がQの中から落とされている。これらにより大幅な計算回数の削減が可能となる。

【００２９】

次に２倍算について考える。前と同様に従来技術の手続き１（２）を変更すると、次の変形２を得る。

変形２

Input　a_1, b_1

Output　a', b'

$a_3(x)=a_1^2$

$b_3(x)=(b_1^2+f) \mod a_3$

$a_4(x)=(f+b_3+b_3^2)/a_3$

$a_4(x)$のモニック化（最高次数の項の係数を１にする）

$b_4(x)=(b_3+1) \mod a_4$

while (deg $a_4 > g$) {

$a'=a_5(x)=(f+b_4+b_4^2)/a_4$

$b'=b_5(x)=(b_4+1) \mod a_5$

$a_4=a'$, $b_4=b'$

}

end

【００３０】

通常の加算の場合と同様に、a_4の計算で被除多項式は１０次式となり、多くの計算が必要となる。これを減らすために、q（x）＝Q（b_3, a_1）を導入する。

補題２

変形２の$a_4(x)$はq（x）を用いて、$a_4(x)=q^2+Q(f, a_3)$で与えられる。

（証明）

deg b_3<deg a_3よりQ（b_3, a_3）＝０であるから、

$a_4(x)=Q(f, a_3)+Q(b_3^2, a_3)$

である。第２項はQ（b_3^2, a_3）＝Q（b_3, a_3）^{2}となる。これは、$b_3=r_1+s_1/a_1$, deg s_1<deg a_1とすると標数２の体の上では、$b_3=r_1^2+s_1^2/a_1^2$, deg s_1^2<deg a_1^2が成り立つことからも明らかである。ゆえに、$a_4(x)=q^2+Q(f, a_3)$　Q．E．D．

【００３１】

加算と同様にf（x）＝x^7を用いると、$a_3(x)=a_1^2(x)$の中に奇数次の項がないことからQ（x^7, a_3）＝xとなる。これよりq(x)を用いて変形２を書き直すと次の本発明のアルゴリズム（２倍算）を得る。

本発明のアルゴリズム（２倍算）

Input　a_1, b_1

Output　a', b'

$b_3(x)=b_1^2+x(a_1-x^3)^2$

$q(x)=Q(b_3, a_1)$

$a_4(x)=q^2+Q(f, a_3)$

a_4をモニック化（最高次数の項の係数を１にする）

$b_4(x)=(b_3+1) \bmod a_4$

if (deg $a_4>g$) then

$a'=a_5(x)=Q(x^7+b_4^2, a_4)$

$b'=b_5(x)=(b_4+1) \bmod a_5$

else　$a'=a_4$, $b'=b_4$

end

【００３２】

なお$b_3(x)$の計算おいては、$x^7 \bmod a_3=x(x^3)^2 \bmod a_3=x(a_1-(a_1-x^3))^2 \bmod a_3=x(a_1-x_3)^2 \bmod a_3=x(a_1-x_3)^2$、deg $b_1{}^2<$deg a_3であることから$b_1^2 \bmod a_3=b_1^2$となることを用いている。また、$b_3(x)$の計算結果をストアしておく必要はない。なぜなら標数２のガロア体上で２乗器は小さなハードウエアで実現できるので、レジスタの代わりに２乗器を持つ方がサイズ的に有利だからである。特に正規基底を用いる場合にはビットシフトだけで実現できる。$b_3(x)$が必要な時は、$a_1(x)$, $b_1(x)$を２乗器に入力し、その出力を直接用いればよい。

【００３３】

通常の加算と同様に、このアルゴリズムでは、次数の高い$a_3(x)$が消えて、その剰余演算及び除算を行う必要がなくなっている。また$a_4(x)$の次数は４次で標数が２であることより、それを計算するのに２乗算のみ必要で乗算は必要ない。さらに$a_5(x)$の計算においても、必要のない$b_4(x)$がQの中から落とされている。

【００３４】

なお補題１及び２はｈ(ｘ)＝１以外でも成り立つ。また、Ｑ（ｆ，a_3）はｆの次数が２ｇ＋１次、a_3の次数が２ｇ次であることに注意すると容易に計算することができる。

【００３５】

また、超楕円曲線は上で使った$y^2+y=x^7$以外のものであってもよい。例えば、ｇ＝３の場合には、Ｋ＝ＧＦ（2^{61}）ｆ（ｘ）＝x^7＋ｘ＋１や、Ｋ＝ＧＦ（2^{67}）ｆ（ｘ）＝x^7＋１等がある。上のアルゴリズムでx^7としていたところを、このようなｆ（ｘ）に置き換えれば、新たにｑ（ｘ）を導入する効果はある。

【００３６】

では、図１に上記アルゴリズムの実装例を示す。レジスタ群１はセレクタ１（３）とセレクタ２（９）に接続されている。セレクタ１（３）及びセレクタ２（９）は共に乗算器及び二乗器５とインバータ７に接続されている。セレクタ１（３）はレジスタ入力用のセレクタであり、セレクタ２（９）は乗算器及び二乗器及びインバータ入力用のセレクタである。なお、セレクタ１（３）とセレクタ２（９）と乗算器及び二乗器５とインバータ７は、コントローラ１１によりそれらの動作が制御されている（図１の点線表示）。レジスタ群１は、ワークエリアとして且つ結果格納のために用いられるレジスタＵreg、Ｘreg、Ｙreg及びＺregと、入力となる$a_1(x)$，$a_2(x)$，$b_1(x)$及び$b_2(x)$を各々格納するレジスタa_1，a_2，b_1及びb_2を含む。なお、ＵregとＸregの記憶位置は４つあり、残りのレジスタの記憶位置は３つである。また、図示していないが、加算器は乗算器及び二乗器５等に設けられており、コントローラ１１により加算が指示されている場合には加算器を動作させる。

【００３７】

では、本発明のアルゴリズム（通常の加算）を実施する際に、図１の回路がどのように動作するかを説明する。図２にレジスタ群１の初期状態を示す。前提として、$s_1(x)=s_{12}x^2+s_{11}x+s_{10}$の各係数をＹregが格納している。また、$a_1$，$a_2$，$b_1$及び$b_2$は、それぞれ$a_1(x)$，$a_2(x)$，$b_1(x)$，$b_2(x)$の係数を格

納している。但し、最高次である３次の項の係数は１であるから、これらは格納する必要がない。すなわち、a_1はc_2，c_1及びc_0を、a_2はe_2，e_1及びe_0、b_1はd_2，d_1及びd_0、b_2はf_2，f_1及びf_0を格納している。

【００３８】

まず$q(x)=s_1(b_1+b_2) \bmod a_2$を得るための計算を実施する。セレクタ２（９）は、以下の計算を実施すべく、レジスタ群１から必要な値を取り出し、乗算器及び二乗器５にそれらを入力する。

（１）$p_4=(s_{12}\cdot b'_2)$ [x^4の係数]

$p_3=(s_{12}\cdot b'_1+s_{11}\cdot b'_2)$ [x^3の係数]

$p_2=s_{12}\cdot b'_0$ [x^2の係数]

但し、

(b_1+b_2)

$=(d_2+f_2)x^2+(d_1+f_1)x+(d_0+f_0)$

$=b'_2x^2+b'_1x+b'_0$

としている。

【００３９】

なお、

$s_1(b_1+b_2)$

$=(s_{12}\cdot b'_2)x^4$

$+(s_{12}\cdot b'_1+s_{11}\cdot b'_2)x^3$

$+(s_{12}\cdot b'_0+s_{11}\cdot b'_1+s_{10}\cdot b'_2)x^2$

$+(s_{11}\cdot b'_0+s_{10}\cdot b'_1)x$

$+s_{10}\cdot b'_0$

である。

よって（１）の計算は、$s_1(b_1+b_2)$の４次及び３次の項の完全な係数と、２次の項の一部の係数の、計算である。これらの計算結果はセレクタ１（３）によりUregに格納される（図３：Uregのみ図示）。（１）のような計算をするのは、乗算器及び二乗器５の乗算器が一度に４つしか使えないという前提があるか

らで、Uregは４つの格納場所を有しているため、乗算器の数に制限が無ければ上位４つの項の係数を（１）で計算することも可能である。また、a_2の剰余計算を行うので、$s_1(b_1+b_2)$の、a_2の最高次の次数である３未満の次数の項はそのまま残ってしまう。よって、２次以下の項の係数を、剰余計算の後から加算するようにしても結果は同じである。

【００４０】

次に、$p_4x^4+p_3x^3+p_2x^2$に対して$a_2(x)=x^3+e_2x^2+e_1x+e_0$の剰余計算及び$s_1(b_1+b_2)$の０次の項の係数の計算$s_{10}\cdot b'_0=p_0$を行う。よって、セレクタ２（９）は必要な値を取り出し、乗算器及び二乗器５にそれらを入力する。

（２）$(p_4x^4+p_3x^3+p_2x^2) \mod a_2$

$s_{10}\cdot b'_0$［x^0の係数］

【００４１】

$(p_4x^4+p_3x^3+p_2x^2) \mod a_2$をより詳しく記述すると以下のようになる。

$p'_3=(p_3+p_4\cdot e_2)$［x^3の係数］

$p'_2=(p_2+p_4\cdot e_1)$［x_2の係数］

$p'_1=p_4\cdot e_0$［xの係数］

そして、

$p'_0=s_{10}\cdot b'_0$［x^0の係数］

も実施する。

これらの計算結果をセレクタ１（３）はUregに格納する（図４：Uregのみ図示）。（２）のような計算を実施するのも乗算器及び二乗器５の乗算器の数が４つになっているからである。

【００４２】

次に$p'_3x^3+p'_2x^2+p'_1x+p'_0 \mod a_2$の計算を行う。これをさらに詳しく記述すると以下のようになるので、セレクタ２（９）は必要な値を取り出し

、乗算器及び二乗器５にそれらを入力する。

（３）$p''_2=(p'_2+p'_3\cdot e_2)$ ［x^2の係数］

$p''_1=(p'_1+p'_3\cdot e_1)$ ［x^1の係数］

$p''_0=(p'_0+p'_3\cdot e_0)$ ［x^0の係数］

これらの計算結果をセレクタ１（３）はUregに格納する（図５：Uregのみ図示）。

【００４３】

以上の計算では$s_1(b_1+b_2)$のうち、$(s_{11}\cdot b'_1+s_{10}\cdot b'_2)x^2+(s_{11}\cdot b'_0+s_{11}\cdot b'_1)x$について考慮されていない。よって、以下のような計算をすべく、セレクタ２（９）は必要な値を取り出し、乗算器及び二乗器５にそれらを入力する。

（４）$p''_2+(s_{11}\cdot b'_1+s_{10}\cdot b'_2)$ ［x^2の係数］

$p''_1+(s_{11}\cdot b'_0+s_{10}\cdot b'_1)$ ［xの係数］

これらの計算にて$q(x)=q_2x^2+q_1x+q_0$は求まった。セレクタ１（３）は、これらの計算結果をZregに格納する（図６：Zregのみ図示）。

【００４４】

次に$a_4(x)=Q(q^2a_1,\ a_2)+x+c_2+e_2$の計算を実施する。そのため、最初に$q^2a_1$の計算を行う。但し、$a_2$の商の計算であるから２次以下の項は計算する必要が無い。以下の計算を行うため、セレクタ２（９）は、必要な値をレジスタ群１から取り出し、乗算器及び二乗器５にそれらを入力する。

（１）$p_7=q_2^2$ ［x^7の係数］

$p_6=q_2^2\cdot c_2$ ［x^6の係数］

$p_5=q_2^2\cdot c_1+q_1^2$ ［x^5の係数］

$p_4=q_1^2\cdot c_2+q_2^2\cdot c_0$ ［x^4の係数］

ここではq^2a_1の３次の項の計算は乗算器の数及びUregの記憶位置が足りないので実施しない。これらの計算結果をセレクタ１（３）は、Uregに格納する

（図７：Uregのみ図示）。

【００４５】

なお、q^2a_1は以下のようになっている。

$$q^2a_1=$$
$$q_2^2x^7+$$
$$q_2^2\cdot c_2x^6+$$
$$(q_2^2\cdot c_1+q_1^2)\ x^5+$$
$$(q_1^2\cdot c_2+q_2^2\cdot c_0)\ x^4+$$
$$(q_1^2\cdot c_1+q_0^2)\ x^3+$$
$$(q_1^2\cdot c_0+q_0^2\cdot c_2)\ x^2+$$
$$q_0^2c_1x+q_0^2c_0$$

【００４６】

（１）の計算の開始と共に、q_2^2の逆数の計算を開始する。このため、セレクタ２（９）は、乗算器及び二乗器５で計算されたq_2^2の結果をインバータ７に入力する。$q^-=1/q_2^2$

とする。

【００４７】

次に、a_2による商を求める計算を行う。これはa_2による剰余計算を行っていくことにより行われる。よって、まず以下のような計算を行うため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（２）$p'_6=p_6+p_7\cdot e_2$ [x^6の係数]

$p'_5=p_5+p_7\cdot e_1$ [x^5の係数]

$p'_4=p_4+p_7\cdot e_0$ [x^4の係数]

$p'_3=(q_1^2\cdot c_1+q_0^2)$ [x^3の係数]

a_2による商を計算する場合、p_7x^4が最初に求まる項であるが、p_7は既に求まっており且つ最終的に求めようとしているa_4はモニック化されるので、Uregに格納しておく必要はない。これらの計算結果をセレクタ１（３）は、Uregに

格納する（図８：Uregのみ図示）。

【００４８】

さらに、a_2による剰余計算を行う。但し、この計算においてはa_4の３次の項の係数（モニック化前）も求まるので、合わせてUregに格納することになる。よって、以下のような計算を行うため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（３）$p''_5 = p'_5 + p'_6 \cdot e_2$ [x^5の係数]

$p''_4 = p'_4 + p'_6 \cdot e_1$ [x^4の係数]

$p''_3 = p'_3 + p'_6 \cdot e_0$ [x^3の係数]

なお、モニック化前のa_4を以下のように記載する。

$a_4(x) = a'_{44}x^4 + a'_{43}x^3 + a'_{42}x^2 + a'_{41}x + a'_{40}$

但し、$a'_{43} = p'_6$ [a_4の３次の項の係数] となる。

セレクタ１（３）は、p'_6をUregから取り出し、これらの計算結果と共にUregに格納する（図９：Uregのみ図示）。

【００４９】

さらに、a_2による剰余計算を行う。但し、この計算においてはa_4の２次の項の係数（モニック化前）も求まるので、合わせてUregに格納することになる。よって、以下のような計算を行うため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（４）$p_{34} = p''_4 + p''_5 \cdot e_2$ [x^4の係数]

$p_{33} = p''_3 + p''_5 \cdot e_1$ [x^3の係数]

但し、$a'_{42} = p''_5$ [a_4の２次の項の係数] となる。

セレクタ１（３）は、p''_5及びa'_{43}をUregから取り出し、これらの計算結果と共にUregに格納する（図１０：Uregのみ図示）。

【００５０】

さらに、a_2による剰余計算を行う。但し、この計算においてはa_4の１次の項の係数（モニック化前）も求まるので、合わせてUregに格納することになる。

また、a_4のＱ以外の項の加算も行う。よって、以下のような計算を行うため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（５）$p_{43}=p_{33}+p_{34}\cdot e_2$［$x^3$の係数］

$a'_{41}=p_{34}+1$［a_4の１次の項の係数］

セレクタ１（３）は、a'_{42}及びa'_{43}をUregから取り出し、これらの計算結果と共にUregに格納する（図１１：Uregのみ図示）。

【００５１】

次に、a_4の０次の項の係数（モニック化前）を計算するため及びa_4のＱ以外の項の加算も行うため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（６）$a'_{40}=p_{43}+c_2+e_2$［a_4の定数項］

セレクタ１（３）は、a'_{42}、a'_{43}及びa'_{41}をUregから取り出し、これらの計算結果と共にUregに格納する（図１２：Uregのみ図示）。これにてa_4のモニック化前の値が求まった。

【００５２】

次に、a_4のモニック化を行う。a_4は４次であり、その係数はq_2^2である。よって、段落番号００４６で説明した計算の終了を待って、q^-をUregの各係数に掛ける。すなわち、以下の計算を行うため、セレクタ２（９）はインバータ７とレジスタ群１から必要な値を取り出し、乗算器及び二乗器５に入力する。

（７）$a'_{43}\cdot q^-$

$a'_{42}\cdot q^-$

$a'_{41}\cdot q^-$

$a'_{40}\cdot q^-$

セレクタ１（３）は、これらの計算結果をXregに格納する（図１３：Xregのみ図示）。これにてモニック化されたa_4が求まった。

【００５３】

次に$b_4(x)=(q\cdot a_1+b_1+1) \bmod a_4$を計算する。まず、$(q\cdot a_1+b_1+1)$の計算を行うが、Uregの記憶位置の数及び乗算器の数の制限のため、以下のように計算する。なお、a_4は４次の式であるから、$(q\cdot a_1+b_1+1)$の３次以下の項は、剰余計算の後に加算しても結果は同じである。セレクタ２（９）は必要な値をレジスタ群１から取り出して、乗算器及び二乗器５に入力する。

（８）$p_5=q_2$［x^5の係数］

$p_4=(q_2\cdot c_2+q_1)$［x^4の係数］

$p_3=(q_2\cdot c_1+q_1\cdot c_2+q_0)$［$x^3$の係数］

$p_2=(d_1+q_1\cdot c_0)$［x_2の係数］

セレクタ１（３）は、これらの計算結果をUregに格納する（図１４：Uregのみ図示）。

【００５４】

なお、

$q\cdot a_1+b_1+1=$

q_2x^5+

$(q_2c_2+q_1)\ x^4+$

$(q_2c_1+q_1c_2+q_0)\ x^3+$

$(d_2+q_2c_0+q_1c_1+q_0c_2)\ x^2+$

$(d_1+q_1c_0+q_0c_1)\ x+$

$d_0+q_0c_0+1$

【００５５】

そして、a_4による剰余を計算する。なお、この剰余計算によりx^1の項が出現するので、d_1xの加算も行う。詳細に記述すると以下のような計算を実行する。このため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（９）$p'_4=p_4+p_5\cdot a_{43}$［x^4の係数］

$p'_3 = p_3 + p_5 \cdot a_{42}$ [x^3の係数]
$p'_2 = p_2 + p_5 \cdot a_{41}$ [x^2の係数]
$p'_1 = p_1 + p_5 \cdot a_{40} + d_1$ [x^1の係数]

セレクタ１（３）は、これらの計算結果をUregに格納する（図１５：Uregのみ図示）。

【００５６】

再度a_4による剰余計算を行う。なお、この剰余計算によりx^0の項の係数が計算されるので、d_0+1の加算も行う。詳細に記述すると以下のような計算を実行する。このため、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（１０）$p''_3 = p'_3 + p'_4 \cdot a_{43}$ [x^3の係数]
$p''_2 = p'_2 + p'_4 \cdot a_{42}$ [x^2の係数]
$p''_1 = p'_1 + p'_4 \cdot a_{41}$ [x^1の係数]
$p''_0 = p'_4 \cdot a_{40} + d_0 + 1$ [x^0の係数]

セレクタ１（３）は、これらの計算結果をUregに格納する（図１６：Uregのみ図示）。

【００５７】

次に（$q \cdot a_1 + b_1 + 1$）のうちa_4の剰余計算に影響がなく且つ（８）乃至（１０）でまだ加算されていない項を加算する。より詳細に記述すると、以下のような計算を実施する。セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（１１）$p_{32} = p''_2 + c_1 \cdot q_1$ [x^2の係数]
$p_{31} = p''_1 + c_0 \cdot q_1$ [x^1の係数]

セレクタ１（３）は、これらの計算結果をUregに格納する（図１７：Uregのみ図示）。

【００５８】

（$q \cdot a_1 + b_1 + 1$）のうちa_4の剰余計算に影響がなく且つ（８）乃至（１１）でまだ加算されていない項を加算する。より詳細に記述すると、以下のような計算を実施する。セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（１２）$b_{42} = p_{32} + c_2 \cdot q_0$ [x^2の係数]

$b_{41} = p_{31} + c_1 \cdot q_0$ [x^1の係数]

$b_{40} = p''_0 + c_0 \cdot q_0$ [x^0の係数]

セレクタ１（３）は、これらの計算結果をUregに格納する（図１８：Uregのみ図示）。

【００５９】

このようにして、$b_4(x)$が求められる。なお、$b_4(x) = b_{43}x_3 + b_{42}x^2 + b_{41}x + b_{40}$と記載するものとする。最後に、セレクタ１（３）はUregの内容をYregとZregとに格納する（図１９：YregとZregのみ図示）。

【００６０】

次に、$a_5(x) = Q(x^7 + b_4^2, a_4)$を計算する。$a_4$は４次式であるから、Qの計算においては$x^7 + b_4^2$の３次以下の項は不要となる。$b_4^2 = b_{43}^2 x^6 + b_{42}^2 x^4 + b_{41}^2 x^2 + b_{40}^2$であるから、$b_{43}^2 x^6 + b_{42}^2 x^4 + x^7$のみが使用されることになる。すなわち、以下のような計算を実行するため、セレクタ２（９）は、レジスタ群１から必要な値を取り出し、乗算器及び二乗器５に入力する。

（１）$p_{17} = 1$ [x^7の係数]

$p_{16} = b_{43}^2$ [x^6の係数]

$p_{15} = 0$ [x^5の係数]

$p_{14} = b_{42}^2$ [x^4の係数]

セレクタ１（３）は、これらの計算結果をUregに格納する（図２０：Uregのみ図示）。

【００６１】

次に、a_4による剰余計算を行う。より具体的には、以下の計算を実行する。

よって、セレクタ２（９）は、レジスタ群１から必要な値を取り出し、乗算器及び二乗器５に入力する。

（２－１）

$$p_{26}=p_{16}+p_{17}\cdot a_{43}$$
$$=p_{16}+a_{43}\ [x^{6}\text{の係数}]$$
$$p_{25}=p_{15}+p_{17}\cdot a_{42}$$
$$=a_{42}\ [x^{5}\text{の係数}]$$
$$p_{24}=p_{14}+p_{17}\cdot a_{41}$$
$$=p_{14}+a_{41}\ [x^{4}\text{の係数}]$$

なお、$a_{53}=p_{17}$［a_5の３次の項の係数］となる。

セレクタ１（３）は、$p_{17}=1$を取り出し、これらの計算結果と共にUregに格納する（図２１：Uregのみ図示）。なお、$a_5(x)=a_{53}x^3+a_{52}x^2+a_{51}x+a_{50}$とする。

【００６２】

さらに、a_4による剰余計算を行う。より具体的には、以下の計算を実行する。よって、セレクタ２（９）は、レジスタ群１から必要な値を取り出し、乗算器及び二乗器５に入力する。

（２－２）

$$p_{35}=p_{25}+p_{26}\cdot a_{43}\ [x^{5}\text{の係数}]$$
$$p_{34}=p_{24}+p_{26}\cdot a_{42}\ [x^{4}\text{の係数}]$$

なお、$a_{52}=p_{26}$［a_5の２次の項の係数］となる。

セレクタ１（３）は、p_{17}及びp_{26}を取り出し、これらの計算結果と共にUregに格納する（図２２：Uregのみ図示）。

【００６３】

さらに、a_4による剰余計算を行う。より具体的には、以下の計算を実行する

。よって、セレクタ２（９）は、レジスタ群１から必要な値を取り出し、乗算器及び二乗器５に入力する。

（２－３）

$a_{50}=p_{34}+p_{35}\cdot a_{43}$ ［a_5の定数項］

なお、$a_{51}=p_{35}$［a_5の１次の項の係数］となる。

セレクタ１（３）は、p_{17}及びp_{26}及びp_{35}を取り出し、これらの計算結果と共にUregに格納する（図２３：Uregのみ図示）。これにてa_5が計算できた。

【００６４】

（３）の処理では、セレクタ１（３）が、Uregに格納された$a_5(x)$をXregに格納する（図２４：Xregのみ図示）。

【００６５】

次に、$b_5(x)=(b_4+1) \bmod a_5(x)$を計算する。$b_4$はYregとZregに格納されている。最初に（４）の処理として、セレクタ１（３）がb_{43}，b_{42}，b_{41}及び$b_{40}+1$をUregに格納する（図２５：Uregのみ図示）。

【００６６】

次に、a_5による剰余計算を行う。以下に必要な計算を詳細に記述する。よって、セレクタ２（９）は、必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（５）$b_{52}=b_{42}+b_{43}\cdot a_{52}$［$b_5$の２次の項の係数］

$b_{51}=b_{41}+b_{43}\cdot a_{51}$［$b_5$の１次の項の係数］

$b_{50}=b_{40}+b_{43}\cdot a_{50}$［$b_5$の０次の項の係数］

$b_5(x)=b_{52}x^2+b_{51}x+b_{50}$と表すものとする。

セレクタ１（３）は、これらの計算結果をZregに格納する（図２６：Zregのみ図示）。よって、Xreg及びZregにa_5とb_5が格納されることになる。なお、解として$a'=a_5$、$b'=b_5$である。

【００６７】

では、本発明のアルゴリズム（２倍算）を実施する場合の図１の回路の動作を説明する。図２の初期状態は、２倍算の場合もあまり変わらない。但し、レジスタa_2及びb_2は空となる。

【００６８】

最初にｑ（ｘ）＝Ｑ（b_3，a_1）を計算するため、$b_3(x)=b_1^2+x\cdot(a_1-x^3)^2$を計算する。但し、$a_1$は３次式であるから、$b_3(x)$の３次以上の項を計算すればよい。より詳細に記述すると以下のようになる。セレクタ２（９）は、必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（１）$b_{35}=c_2^2$［x^5の係数］

$b_{34}=d_2^2$［x^4の係数］

$b_{33}=c_1^2$［x^3の係数］

これらの計算結果をセレクタ１（３）はUregに格納する（図２７：Uregのみ図示）。

【００６９】

なお、

$b_1^2+x\cdot(a_1-x^3)^2=$

$c_2^2x^5+d_2^2x^4+c_1^2x^3+d_1^2x^2+c_0^2x+d_0^2$

$=b_{35}x^5+b_{34}x^4+b_{33}x^3+b_{32}x^2+b_{31}x+b_{30}$

である。

【００７０】

次に、Ｑ（b_3，a_1）を計算する。より詳細に記述すると以下のようになる。セレクタ２（９）は、必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（２－１）

$p_{14}=b_{34}+b_{35}\cdot c_2$［$x^4$の係数］

$p_{13}=b_{33}+b_{35}\cdot c_1$［$x^3$の係数］

$p_{12}=b_{35}\cdot c_0$［x^2の係数］

なお、$q_2 = b_{35}$である。$q(x) = q_2x^2 + q_1x + q_0$と記載する。

セレクタ１（３）は、b_{35}をレジスタ群１から取り出し、これらの計算結果と共に、Uregに格納する（図２８：Uregのみ図示）。

【００７１】

同様に、a_1による剰余計算を行う。より詳細に記述すると以下のようになる。セレクタ２（９）は、必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（２－２）

$p_{23} = p_{13} + p_{14} \cdot c_2$ [x^3の係数]

（$p_{22} = p_{12} + p_{14} \cdot c_1$ [x^2の係数]）

なお、$q_1 = p_{14}$である。また、$q_0 = p_{23}$である。

セレクタ１（３）は、q_2及びp_{14}をレジスタ群１から取り出し、これらの計算結果と共に、Uregに格納する（図２９：Uregのみ図示）。

【００７２】

なお、同時にc_2^2の逆数を求めるため、セレクタ２（９）はc_2^2を乗算器及び二乗器５から受け取り、インバータ７に入力する。ここで、$q^- = 1 / c_2^2$とする。

【００７３】

$a_4(x) = q^2(x) + x$をモニック化する必要があるので、以下の計算を実行する。セレクタ２（９）は、必要な値をレジスタ群１及びインバータ７から取り出し、乗算器及び二乗器５に入力する。

（３）$a_{43} = 0$ [x^3の係数]

$a_{42} = q_1^2 \cdot q^{-2}$ [x^2の係数]

$a_{41} = 1 \cdot q^{-2}$ [xの係数]

$a_{40} = q_0^2 \cdot q^{-2}$ [x^0の係数]

これらの計算結果をセレクタ１（３）はXregに格納する（図３０：Xregのみ図示）。なお、$a_{44} = 1$ [x^4の係数] であるからあえて記憶しておく必要はな

い。（$a_4(x)=x^4+a_{43}x^3+a_{42}x^2+a_{41}x+a_{40}$と記載する。

【００７４】

次に、$b_4=(b_3+1) \bmod a_4$を計算する。a_4は４次式であるから（b_3+1）の３次以下の項は剰余計算の後に加算しても計算結果は変わらない。Uregの記憶位置の数及び乗算器の数の制限を考慮して以下のような計算を行う。なお、セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（４）$b_{35}=c_2^2$ [x^5の係数]

$b_{34}=d_2^2$ [x^4の係数]

$b_{33}=c_1^2$ [x^3の係数]

$b_{32}=d_1^2$ [x^2の係数]

これらの計算結果をセレクタ１（３）はUregに格納する（図３１：Uregのみ図示）。

【００７５】

そしてa_4による剰余計算を行う。但し、b_3の１次の項を加算する。より詳細に記述すれば以下のような計算を行う。セレクタ２（９）は必要な値をレジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（５）$p_{14}=b_{34}+b_{35}\cdot a_{43}$

$=b_{34}$ [x^4の係数]

$p_{13}=b_{33}+b_{35}\cdot a_{42}$ [x^3の係数]

$p_{12}=b_{32}+b_{35}\cdot a_{41}$ [x^2の係数]

$p_{11}=b_{35}\cdot a_{40}+c_0^2$ [x^1の係数]

これらの計算結果をセレクタ１（３）はUregに格納する（図３２：Uregのみ図示）。

【００７６】

さらにa_4による剰余計算を行う。但し、b_3の０次の項及び１を加算する。より詳細に記述すれば以下のような計算を行う。セレクタ２（９）は必要な値をレ

ジスタ群１から取り出し、乗算器及び二乗器５に入力する。

（６）$p_{23}=p_{13}+p_{14}\cdot a_{43}$ [x^3の係数]

$p_{22}=p_{12}+p_{14}\cdot a_{42}$ [x^2の係数]

$p_{21}=p_{11}+p_{14}\cdot a_{41}$ [x^1の係数]

$p_{20}=p_{14}\cdot a_{40}+d_0^2+1$ [x^0の係数]

これらの計算結果をセレクタ１（３）はＹregとＺregに格納する（図３３：Ｙreg及びＺregのみ図示）。これにて、$b_4(x)$が求まった。

【００７７】

これ以降の計算は、通常の加算の場合と同じである。

【００７８】

以上の処理を処理フローとして示すと図３４のようになる。最初に、$a_1(x)$, $b_1(x)$, $a_2(x)$, b_2が入力される（ステップ１００）。２倍算の場合には、$a_1(x)$及び$b_1(x)$のみ入力される。次に、通常の加算か２倍算かで処理を切り換える（ステップ１１０）。もし、２倍算の場合には、$q^-=1/c_2^2$を計算する（ステップ１２０）。また、ｑ（ｘ）＝Ｑ（b_3, a_1）をＵregに格納する（ステップ１３０）。図１のような回路ではステップ１２０及び１３０は同時に実行される。そして、モニック化された$a_4(x)=q^{-2}$（$Ureg^2+x$）を計算して、Ｘregに格納する（ステップ１４０）。一方。ステップ１１０で通常の加算であると判断された場合には、a_1及びa_2の最大公約多項式を計算する。最大公約多項式が１でない場合は、本発明では取り扱わない。そして、$s_1(x)a_1(x)+s_2(x)a_2(x)=1$である$s_1$を計算し、Ｙregに格納する（ステップ１５０）。次にｑ（ｘ）＝$s_1(b_1+b_2)$ mod a_2を計算し、Ｚregに格納する（ステップ１６０）。そして、$q^-=1/q_2^2$を計算する（ステップ１７０）。また、Ｑ（q^2a_1, a_2）＋ｘ＋c_2+e_2を計算し、Ｕregに格納する（ステップ１８０）。ステップ１７０及びステップ１８０は図１の回路では同時に実行する。そして、モニック化された$a_4(x)=q^-Ureg$を計算し、Ｘregに格納する（ステップ１９０）。

【００７９】

以下の処理は、通常の加算及び２倍算と共通である。$b_4(x)=(b_3+1)$ mo

d a_4を計算し、Yreg及びZregに格納する（ステップ２００）。但し、通常の加算の場合と２倍算の場合では、b_3の定義が異なる。そして、$a_5(x)=Q(x^7+b_2^2, a_4)$を計算し、Xregに格納する（ステップ２１０）。最後に、$b_5(x)=(b_4+1) \bmod a_5$を計算し、Zregに格納する（ステップ２２０）。

【００８０】

図３４のような処理を通常のコンピュータ（例えば図３５）のためのコンピュータ・プログラムにて実装することができる。但し、二乗算は通常のコンピュータでは高速に実行できないので、処理速度向上には限界がある。

【００８１】

なお、このような本発明のアルゴリズムを実行するような装置及びプログラムを実装して暗号化装置又は復号化装置又はそれらを含む暗号システムを構築することができる。

【００８２】

【効果】

ヤコビ多様体の群演算を少ない演算量にて実現することができた。

【００８３】

また、ヤコビ多様体の群演算を少ないハードウエア量にて実施可能とすることもできた。

【００８４】

[計算量の評価]

本発明のアルゴリズム（通常の加算及び２倍算）の乗算の実行回数について評価する。以下mは乗算器１つが乗算１回を行うことを、Mは複数の乗算器が同時に乗算を１回実行することを表すと定義する。すなわち、mは乗算回数を表すために用いられ、Mは乗算器群が呼び出された回数を表す。またＩは逆元を求める計算１回を表す。以後Ｉ，Ｍ，ｍを用いて計算量を表すものとする。例えば、Ｉ＋２ｍは１回の逆元演算と２回の乗算が行われることを表す。次の表１及び表２は加算と２倍算の計算量をまとめたものである。

【表１】

計算	計算量	呼出回数	時間
GCD	3I+23m	3I+9M	3t(I)+9t(M)
q(x)	15m	4M	4t(M)
$a_4(x)$	I+20m	I+6M	t(I)+t(M)
$b_4(x)$	17m	5M	5t(M)
$a_5(x)$, $b_5(x)$	6m	3M	3t(M)
合計	4I+81m	4I+27M	4t(I)+22t(M)

【表２】

計算	計算量	呼出回数	時間
q(x)	3m	2M	0
$a_4(x)$	I+2m	I+M	t(I)+t(M)
$b_4(x)$	8m	2M	2t(M)
$a_5(x)$, $b_5(x)$	6m	3M	3t(M)
合計	I+19m	I+8M	t(I)+6t(M)

なお表１及び表２の中ではt(Ｉ)は逆元を計算するための時間を、t(M)は乗算を計算する時間を表す。また2^n乗算は１クロック・サイクルで実行できるものとして無視している。

【００８５】

表１（加算）では t(Ｉ) > 5t(M) であるものと仮定している。これにより$a_4(x)$の計算中に$a_4(x)$をモニック化するための逆元の計算を同時に行うことがで

きる。さらに表２（２倍算）では t(Ｉ) > 2t(M)であるものと仮定している。これによりq(x)の計算と$a_4(x)$をモニック化するために必要な逆元の計算を並列に行うことができる。

【００８６】

表１及び表２に対し、T. Itoh, S.Tsujii, "A Fast Algorithm for Computing Multiplicative Inverse in GF(2m) Using Normal Bases," Inform. and Comput. , vol.83, No. 1, pp.171-177, (1989)記載の方法（以下、伊東－辻井方法と呼ぶ）でＧＦ（2^{59}）上でt(Ｉ)=8t(M)が成り立つことを用いると、通常の加算の場合、計算量は113mであり、時間は54t(M)である。また、２倍算の場合、計算量は27mであり、時間は14t(M)となる。一方、酒井康行，石塚裕一，櫻井幸一，“安全な超楕円曲線暗号の構成とその実装，” SCIS'98-10.1.B, Jan. 1998（以下、文献１と呼ぶ）で得られた結果は表３のとおりである。

【表３】

	加 算		２倍算	
	乗算	逆元演算	乗算	逆元演算
g＝0	3	1	3	1
g＝3	401	0	265	0
g＝11	17477	0	10437	0

【００８７】

表３と表１及び表２を比べると、通常の加算の場合、計算量で３．５倍、時間で７倍、２倍算の場合、計算量で１０倍、時間で１９倍、本発明のアルゴリズムの方が良い。また７個の乗算器の使用効率は、通常の加算で０．５７２、２倍算で０．４５である。よって本発明のアルゴリズムは従来の手法と比べて効率的に計算が実行されると同時に高い並列度を持っている。

【００８８】

［処理性能の評価］

表１及び表２に基づいて１６０bit程度の整数倍を行うのに必要な時間を計算すると表４となる。なお、２倍算は１６０回、加算は８０回行うものと仮定している。

【表４】

動作周波数	１回の乗算に必要なクロック		
	ケースA t(M)＝59クロック	ケースB t(M)＝8クロック	ケースC t(M)＝1クロック
20MHz	19. 35ms	2. 624ms	0. 328ms
40MHz	9. 68ms	1. 312ms	0. 164ms
80MHz	4. 84ms	0. 656ms	0. 082ms

【００８９】

一方文献１のソフトウェアによる実装では、Alpha 21164（２５０ＭＨｚ）（AlphaはDigital Equipment Corp.の商標）を用い、加算には５００μs、２倍算には５０μs、１６０bitの整数倍には１１８ｍｓの処理時間を要した。この結果と比べると本発明のアルゴリズムをハードウェア実装したものは、動作周波数２０ＭＨｚ、Case Ａの場合で約５倍、Case Ｂの場合で約５０倍、Case Ｃの場合で約３６０倍高速に処理を行う。ＲＳＡ暗号において専用ハードウェアによる計算と動作周波数が１０倍程度異なる汎用のＭＰＵによる計算の処理時間の比が５倍程度であることを考えると、超楕円曲線暗号と本発明のアルゴリズムは実用上ハードウェア実装にかなり適していると言える。

【００９０】

また安全性で同等と考えられる１６０bit鍵楕円曲線暗号では、技術速報，NIKKEI ELECTRONICS，1998.3.23，(no.712) pp. 23によると２０ＭＨｚの動作周波数で署名を行うのに最大３．６ｍｓの時間が、また鳥居直哉，岡田壮一，長谷部高行，“楕円曲線暗号チップの試作，”信学ソサイエティ大，A-7-1，Oct. 1998

によると２７Ｋゲートのハードウェアを２０ＭＨｚの動作周波数で、６０ｍｓの平均処理時間がかかることが報告されている。それらと比べると提案アルゴリズムは同等か、数倍高速に処理を行う。

【００９１】

ここで処理性能と消費電力について楕円曲線暗号（ｇ＝１）と超楕円曲線暗号（ｇ＞１となる任意のｇ）を比較する。超楕円曲線暗号の演算は楕円曲線暗号と比べて複雑である。しかし定義体はおおよそ１／ｇのものを用いることができる。一般に定義体をＧＦ（2^n）とすると乗算器のハードウェア量はｎの２乗に、消費電力もｎの２乗に比例し、演算のスピードは１／｛１－（$\log_n$ ｇ）｝に比例すること。よって、乗算器性能の種数に対する依存性はg^4｛１＋$\log_n$ｇ＋（$\log_n g$）^{2}＋．．．｝となる。それに対し演算量の増加はg^2に比例する。よって漸近的には超楕円曲線暗号のほうがg^2｛１＋$\log_n$ｇ＋．．．｝だけ有利である。またハードウェア実装の点から見ても超楕円曲線暗号の方がｇ倍の並列性を実現できることも利点である。

【００９２】

[ゲートアレーにマッピングした場合の評価]

上の説明では評価を行うためにt(M)と動作周波数を用いた。最大動作周波数を求めるためには具体的に回路設計を行い、半導体のテクノロジーにマッピングを行わなければならない。そこで８ クロック（clock）で乗算器が計算を行う表４のCase Ｂの場合についてＶＨＤＬ（IEEE std 1076 -1987）を用いて設計し、有効チャネル長 Leff ＝ ０．２７μｍのＣＭＯＳゲートアレー・テクノロジー（ＩＢＭ ＣＭＯＳ ５ＳＥ）にマッピングした場合の評価を行った。その結果、レジスタ間最大遅延１２ｎｓ（最大動作周波数８３ＭＨｚに対応する）、ハードウェアサイズ約１４０Ｋセル（cell）の結果を得た。個々のブロックのサイズについては表５に示す。

【表５】

図１のブロック	サイズ（セル）	
乗算器	34265セル	乗算器７個
二乗器	1344セル	二乗器３個
インバータ	27414セル	
レジスタ群	18408セル	
コントローラ	9749セル	５９ビットレジスタ２６個（係数１２個を含む）
セレクタ １	37140セル	
セレクタ ２	17402セル	
合　計	145722セル	

なお、全体のセル数が１４０Ｋセルとなったのは、個々のブロックを結合した後で回路全体のタイミングを最適化することにより約５Ｋセル減らすことにより実現されたものである。これらの動作周波数・サイズはＲＳＡ暗号などの暗号化ＶＬＳＩなどと比べても十分実用的な数字である。なおＧＦ（2^{59}）の原始多項式としてｐ（ｘ）＝$x^{59}+x^{6}+x^{5}+x^{4}+x^{3}+x+1$を用いた。理由はＧＦ（$2^{59}$）には、オプティマル・ノーマル・ベース（Optimal Normal Bases：ＧＦ（2^{n}）の正規基底の中で乗算結果１bitが２ｎ－１個の項の和であらわせるもの）が存在しないこと、また基礎体がＧＦ（２）の時、円分体は偶数次の拡大体にしか存在しないからである。

【図面の簡単な説明】

【図１】

本発明の全体のブロック図である。

【図２】

本発明のアルゴリズム（通常の加算）を実施する際の、レジスタ群１の初期状態を示す図である。

【図３】

q（x）＝s_1（b_1＋b_2） mod a_2の途中結果を格納しているUregの状態を示す図である。

【図４】

q（x）＝s_1（b_1＋b_2） mod a_2の途中結果を格納しているUregの状態を示す図である。

【図５】

q（x）＝s_1（b_1＋b_2） mod a_2の途中結果を格納しているUregの状態を示す図である。

【図６】

q（x）の最終結果を格納しているZregの状態を示す図である。

【図７】

$a_4(x)$＝Q（q^2a_1，a_2）＋x＋c_2＋e_2の途中結果を格納しているUregの状態を示す図である。

【図８】

$a_4(x)$＝Q（q^2a_1，a_2）＋x＋c_2＋e_2の途中結果を格納しているUregの状態を示す図である。

【図９】

$a_4(x)$＝Q（q^2a_1，a_2）＋x＋c_2＋e_2の途中結果を格納しているUregの状態を示す図である。

【図１０】

$a_4(x)$＝Q（q^2a_1，a_2）＋x＋c_2＋e_2の途中結果を格納しているUregの状態を示す図である。

【図１１】

$a_4(x)$＝Q（q^2a_1，a_2）＋x＋c_2＋e_2の途中結果を格納しているUregの状態を示す図である。

【図１２】

$a_4(x)$＝Q（q^2a_1，a_2）＋x＋c_2＋e_2の途中結果を格納しているUregの状態を示す図である。

【図１３】

モニック化された$a_4(x)$の最終結果を格納しているXregの状態を示す図である。

【図１４】

$b_4(x)=(q\cdot a_1+b_1+1) \mod a_4$の途中結果を格納しているUregの状態を示す図である。

【図１５】

$b_4(x)=(q\cdot a_1+b_1+1) \mod a_4$の途中結果を格納しているUregの状態を示す図である。

【図１６】

$b_4(x)=(q\cdot a_1+b_1+1) \mod a_4$の途中結果を格納しているUregの状態を示す図である。

【図１７】

$b_4(x)=(q\cdot a_1+b_1+1) \mod a_4$の途中結果を格納しているUregの状態を示す図である。

【図１８】

$b_4(x)=(q\cdot a_1+b_1+1) \mod a_4$の途中結果を格納しているUregの状態を示す図である。

【図１９】

$b_4(x)$の最終結果を格納しているYregとZregの状態を示す図である。

【図２０】

$a_5(x)=Q(x^7+b_4^2, a_4)$の途中結果を格納しているUregの状態を示す図である。

【図２１】

$a_5(x)=Q(x^7+b_4^2, a_4)$の途中結果を格納しているUregの状態を示す図である。

【図２２】

$a_5(x)=Q(x^7+b_4^2, a_4)$の途中結果を格納しているUregの状態を示す図である。

【図２３】

$a_5(x)=Q(x^7+b_4^2,\ a_4)$の途中結果を格納しているUregの状態を示す図である。

【図２４】

$a_5(x)$の最終結果を格納しているXregの状態を示す図である。

【図２５】

$b_5(x)=(b_4+1)\ \mathrm{mod}\ a_5(x)$の途中結果を格納しているUregの状態を示す図である。

【図２６】

$b_5(x)$の最終結果を格納しているZregの状態を示す図である。

【図２７】

$q(x)=Q(b_3,\ a_1)$の途中結果を格納しているUregの状態を示す図である。

【図２８】

$q(x)=Q(b_3,\ a_1)$の途中結果を格納しているUregの状態を示す図である。

【図２９】

$q(x)$の最終結果を格納しているUregの状態を示す図である。

【図３０】

モニック化された$a_4(x)=q^2(x)+x$を格納しているXregの状態を示す図である。

【図３１】

$b_4=(b_3+1)\ \mathrm{mod}\ a_4$の途中結果を格納しているUregの状態を示す図である。

【図３２】

$b_4=(b_3+1)\ \mathrm{mod}\ a_4$の途中結果を格納しているUregの状態を示す図である。

【図３３】

b_4の最終結果を格納しているYregとZregの状態を示す図である。

【図３４】

本発明のアルゴリズムのフローを示す図である。

【図３５】

通常のコンピュータの構成例を示す図である。

【符号の説明】

１　レジスタ群

３　セレクタ１

５　乗算器及び二乗器

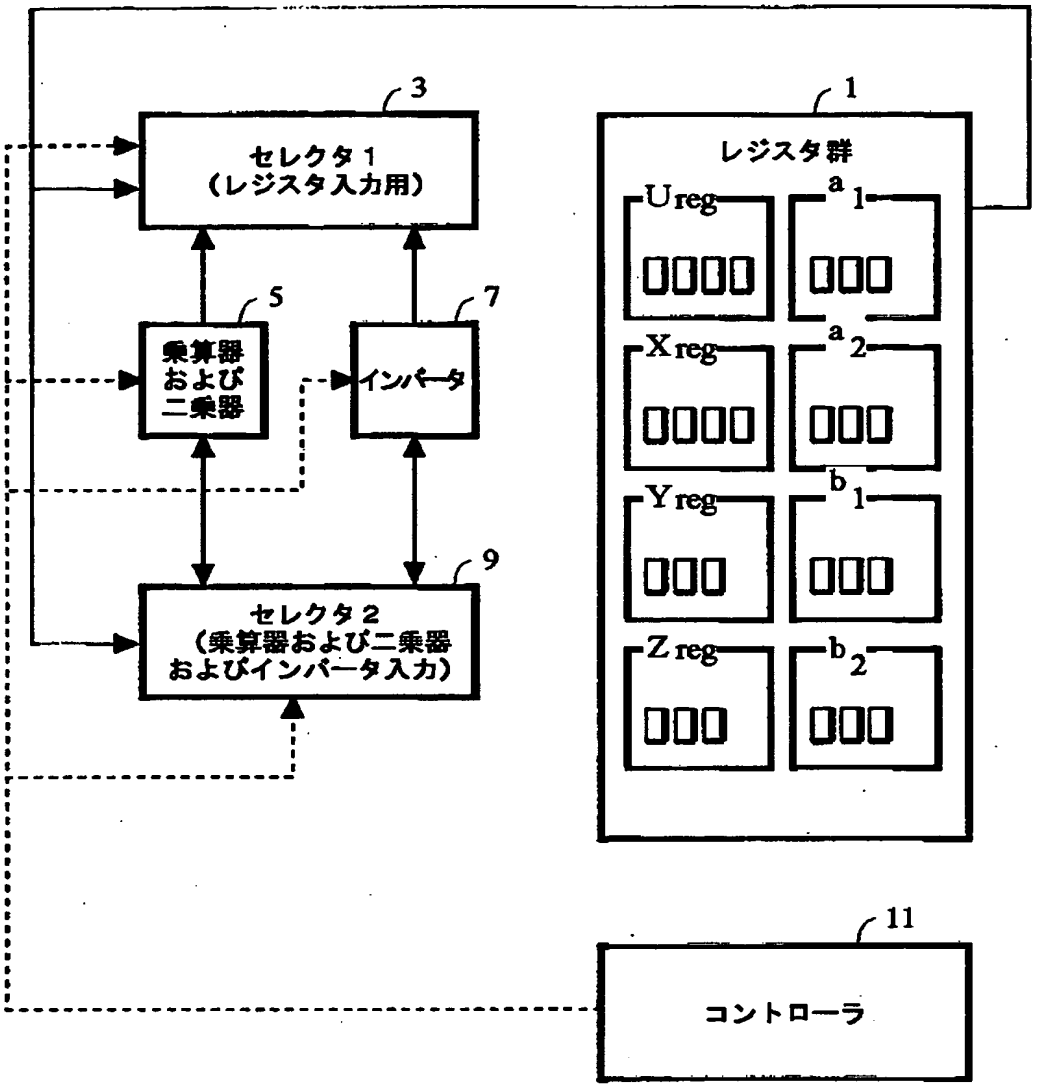
７　インバータ

９　セレクタ２

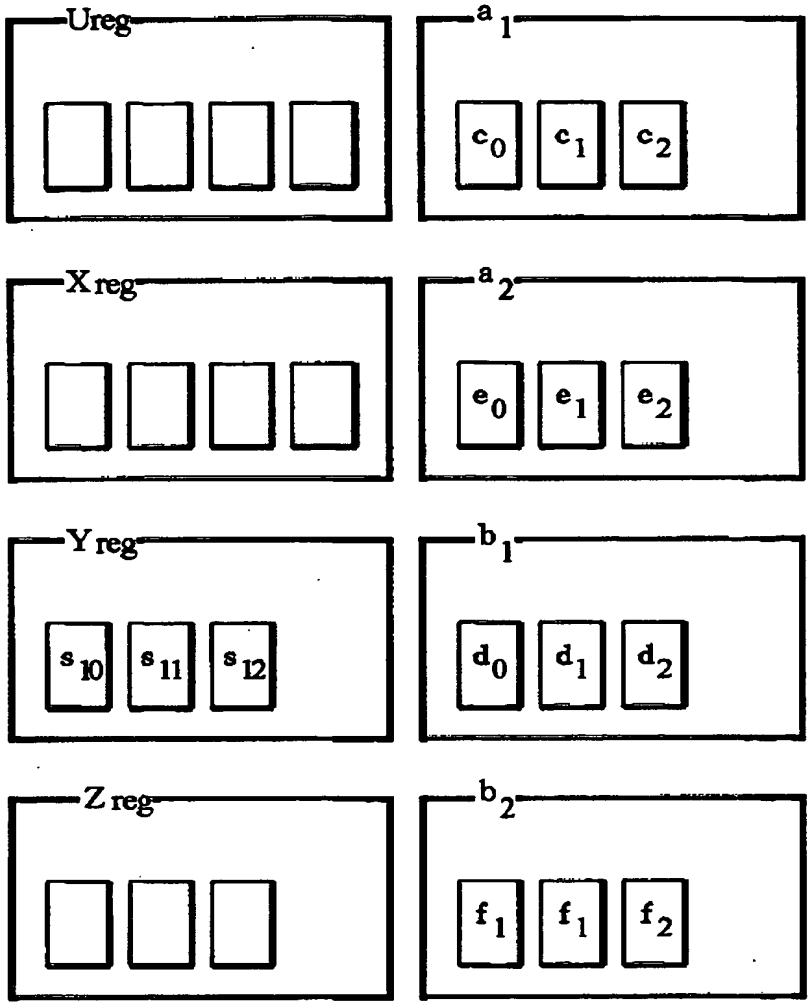
１１　コントローラ

【書類名】　　図面

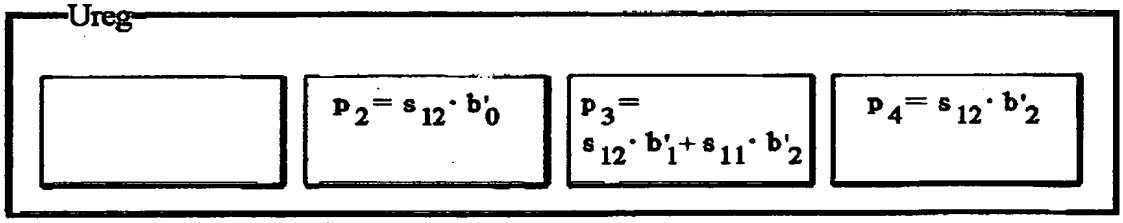
【図１】


3
セレクタ１
（レジスタ入力用）
1
レジスタ群
Ureg
a 1
5
乗算器
および
二乗器
7
インバータ
Xreg
a 2
Yreg
b 1
9
セレクタ２
（乗算器および二乗器
およびインバータ入力）
Zreg
b 2
11
コントローラ

【図２】

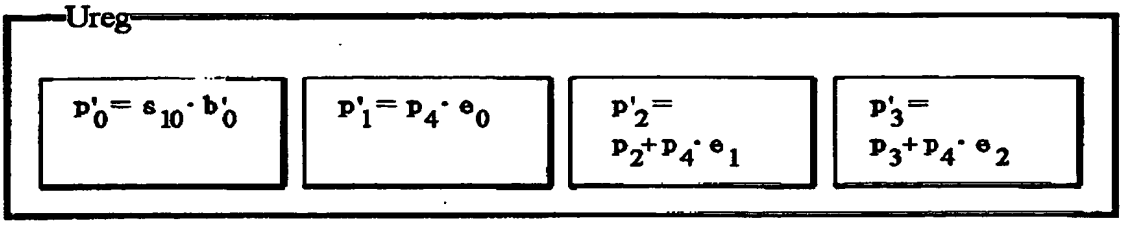


【図３】

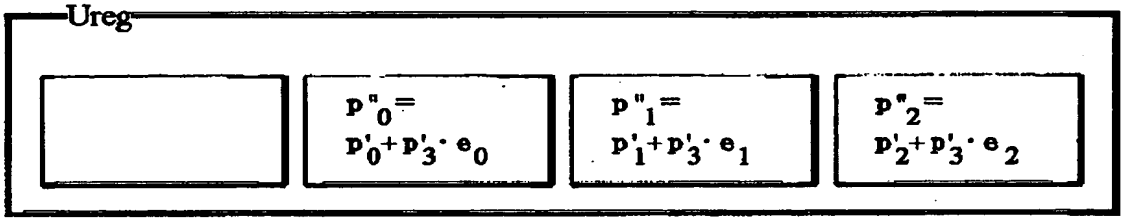


出証特平１１－３０２７５２０

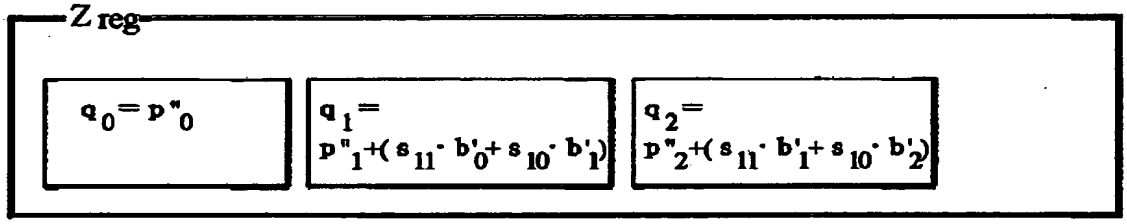
【図４】


Ureg
$p'_0 = s_{10} \cdot b'_0$
$p'_1 = p_4 \cdot e_0$
$p'_2 = p_2 + p_4 \cdot e_1$
$p'_3 = p_3 + p_4 \cdot e_2$

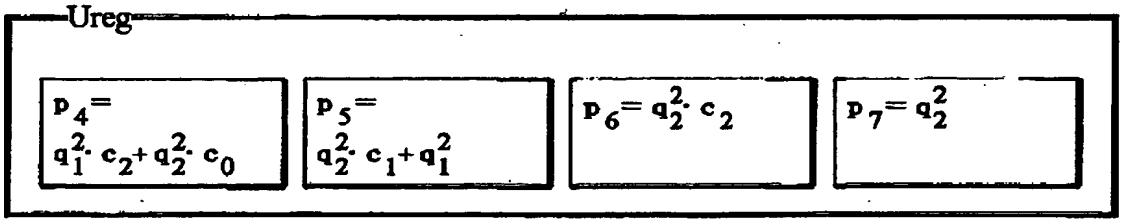
【図５】


Ureg
$p''_0 = p'_0 + p'_3 \cdot e_0$
$p''_1 = p'_1 + p'_3 \cdot e_1$
$p''_2 = p'_2 + p'_3 \cdot e_2$

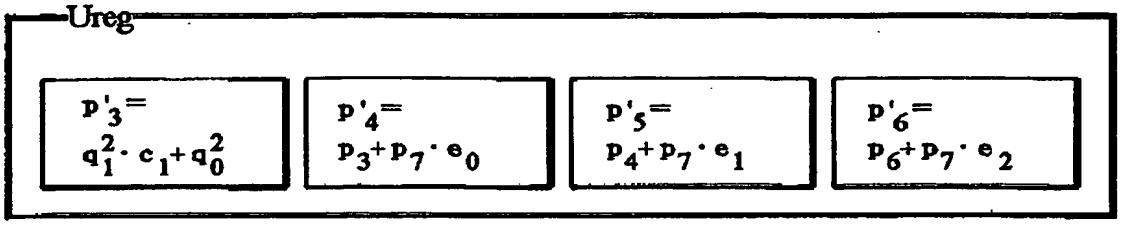
【図６】


Z reg
$q_0 = p''_0$
$q_1 = p''_1 + (s_{11} \cdot b'_0 + s_{10} \cdot b'_1)$
$q_2 = p''_2 + (s_{11} \cdot b'_1 + s_{10} \cdot b'_2)$

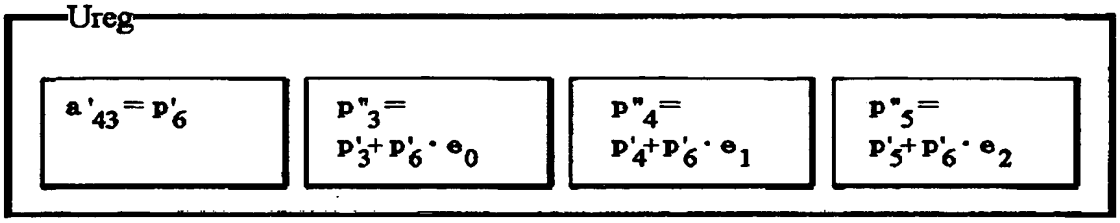
【図７】


Ureg
$p_4 = q_1^2 \cdot c_2 + q_2^2 \cdot c_0$
$p_5 = q_2^2 \cdot c_1 + q_1^2$
$p_6 = q_2^2 \cdot c_2$
$p_7 = q_2^2$

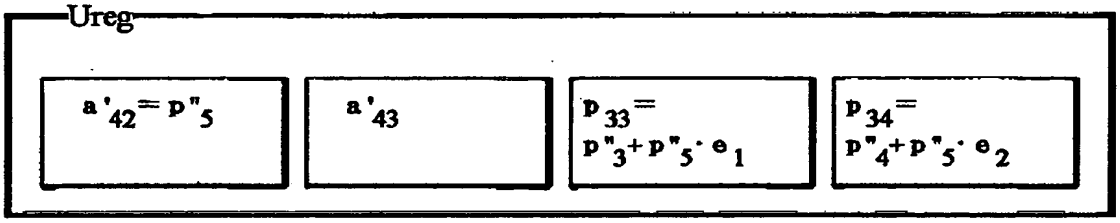
【図８】


Ureg
$p'_3 = q_1^2 \cdot c_1 + q_0^2$
$p'_4 = p_3 + p_7 \cdot e_0$
$p'_5 = p_4 + p_7 \cdot e_1$
$p'_6 = p_6 + p_7 \cdot e_2$

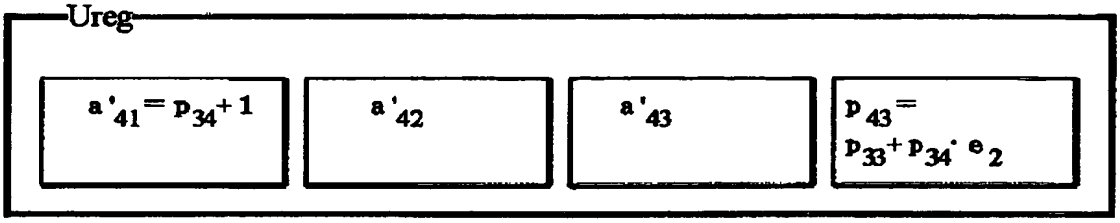
【図９】


Ureg
$a'_{43} = p'_6$
$p''_3 = p'_3 + p'_6 \cdot e_0$
$p''_4 = p'_4 + p'_6 \cdot e_1$
$p''_5 = p'_5 + p'_6 \cdot e_2$

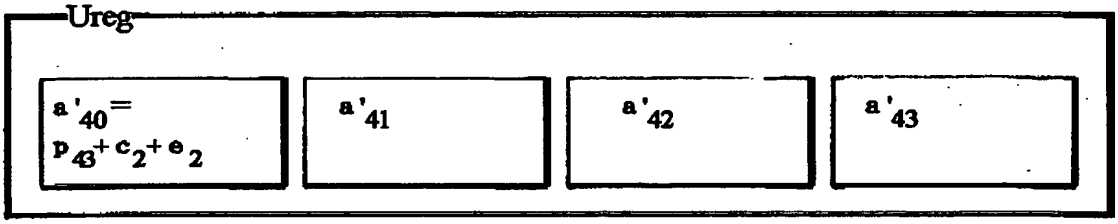
【図１０】


Ureg
$a'_{42} = p''_5$
a'_{43}
$p_{33} = p''_3 + p''_5 \cdot e_1$
$p_{34} = p''_4 + p''_5 \cdot e_2$

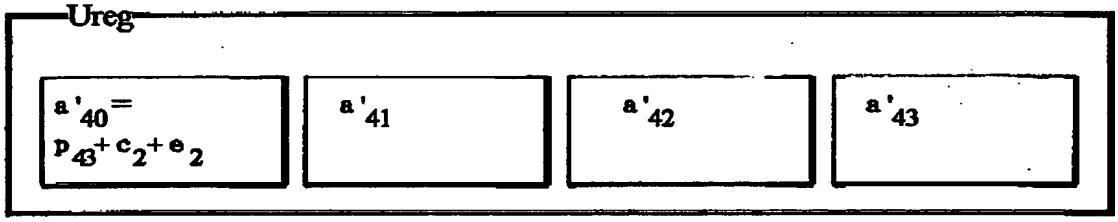
【図１１】


Ureg
$a'_{41} = p_{34} + 1$
a'_{42}
a'_{43}
$p_{43} = p_{33} + p_{34} \cdot e_2$

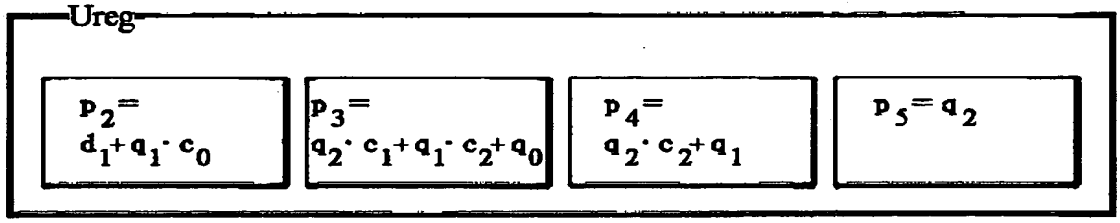
【図１２】



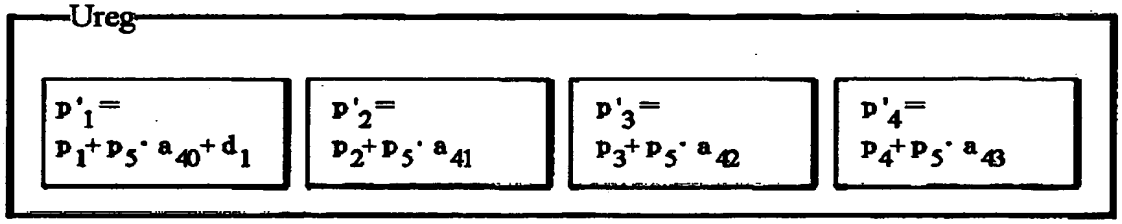
【図１３】



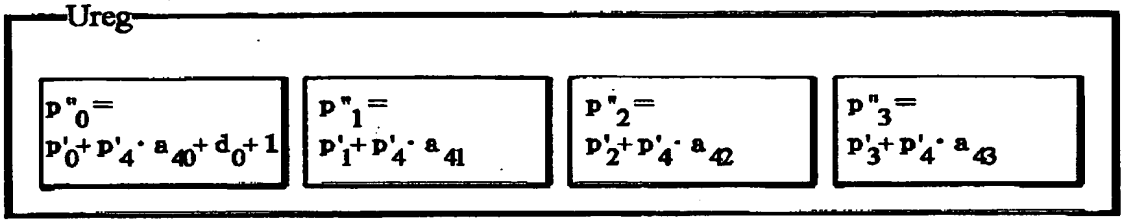
【図１４】



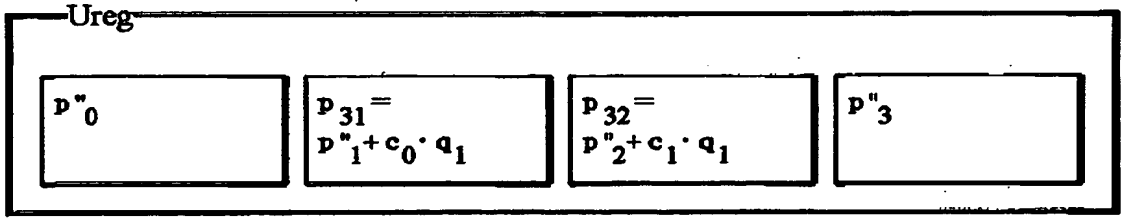
【図１５】



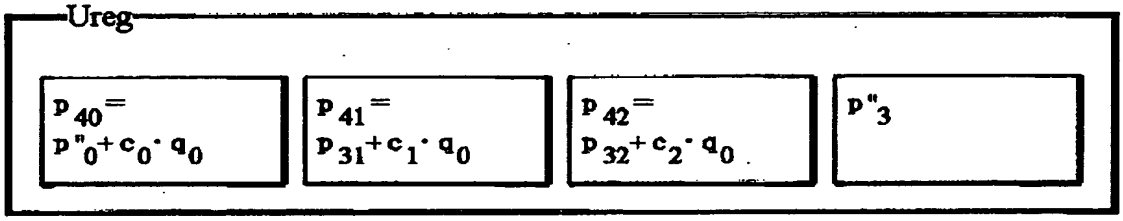
【図１６】



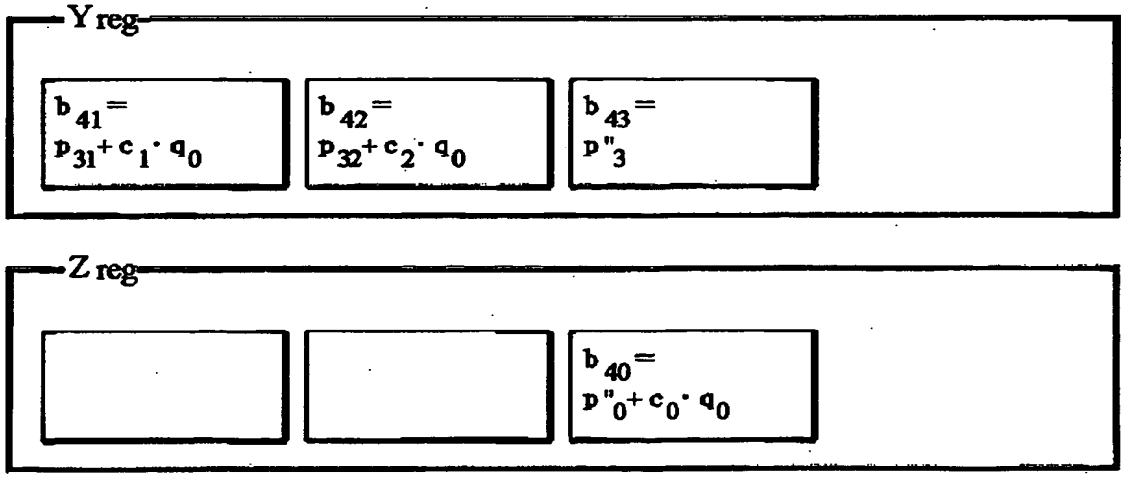
【図１７】



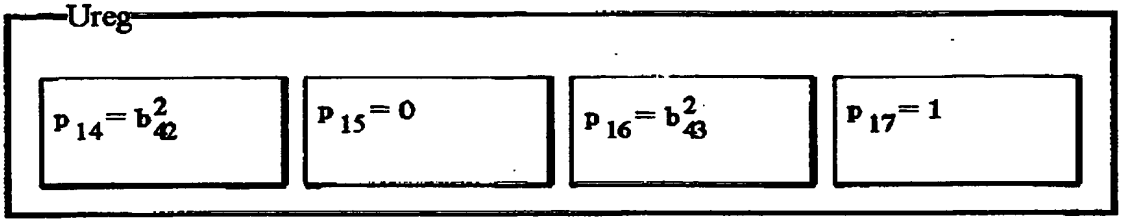
【図１８】



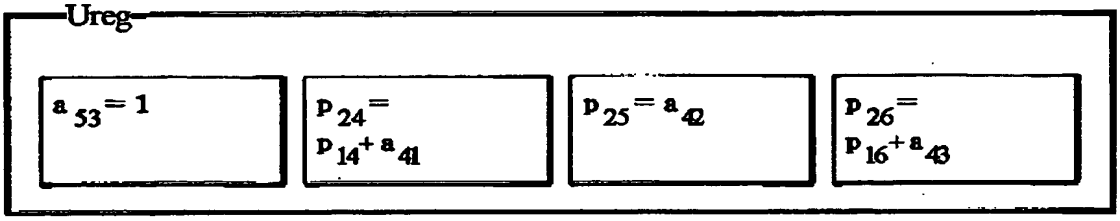
【図１９】



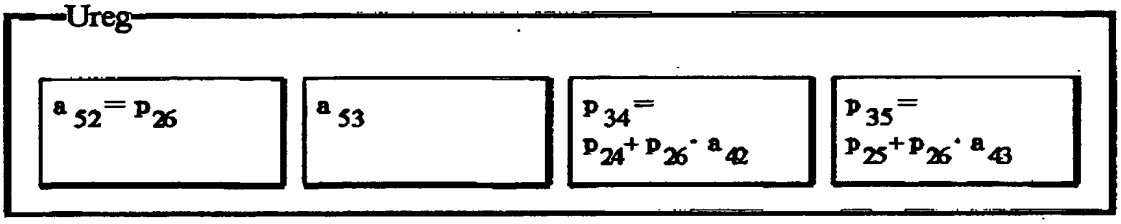
【図２０】



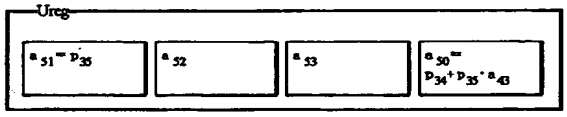
【図２１】



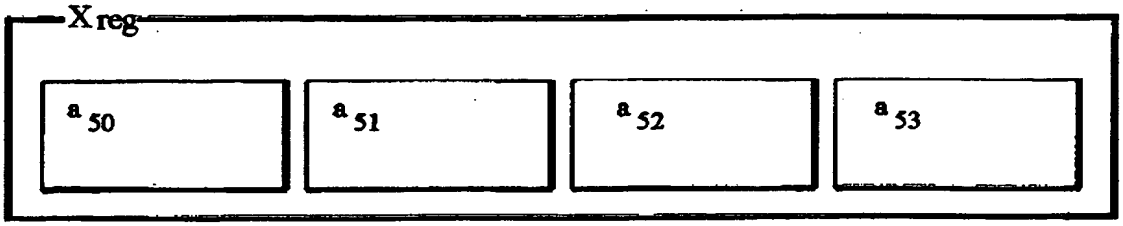
【図２２】



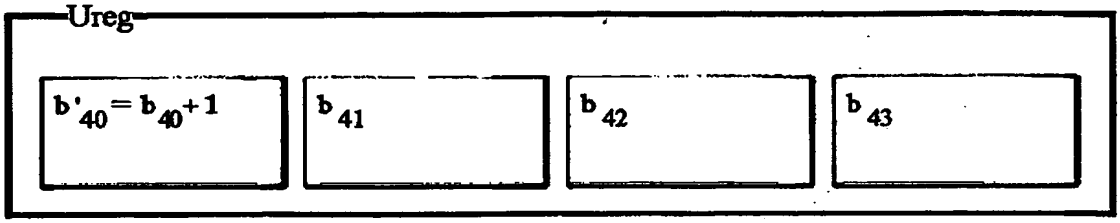
【図２３】



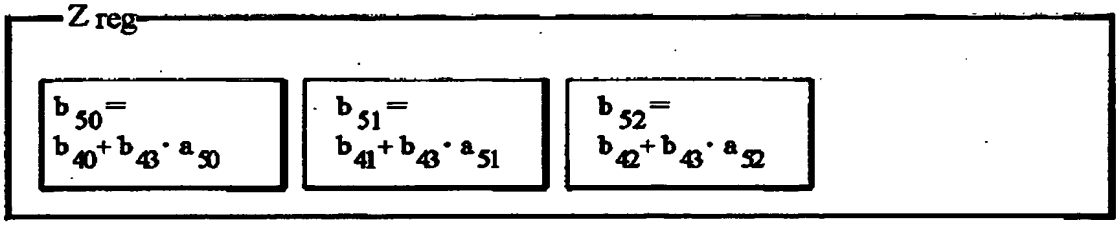
【図２４】



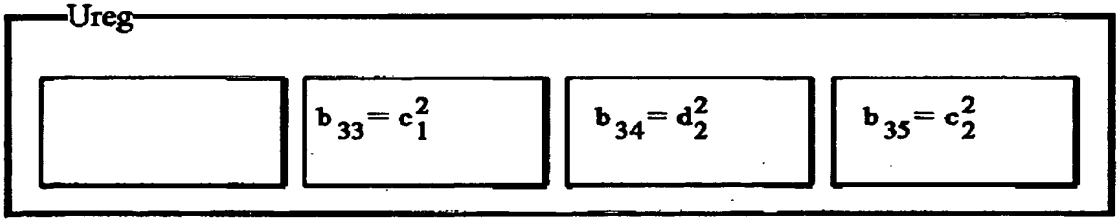
【図２５】



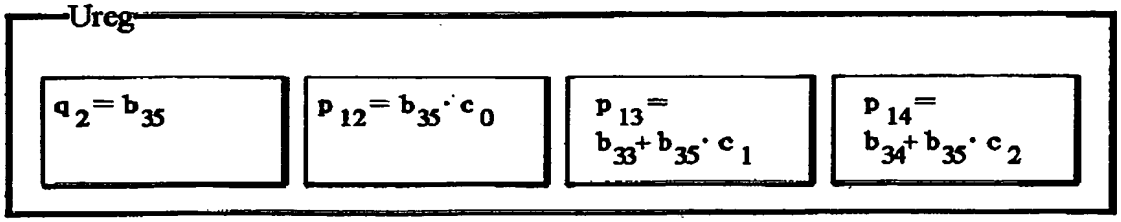
【図２６】



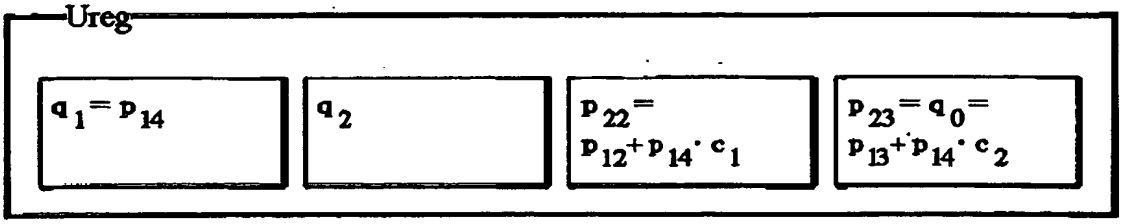
【図２７】



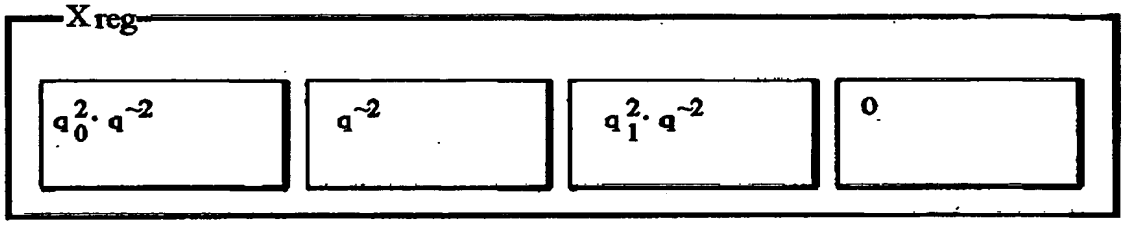
【図２８】



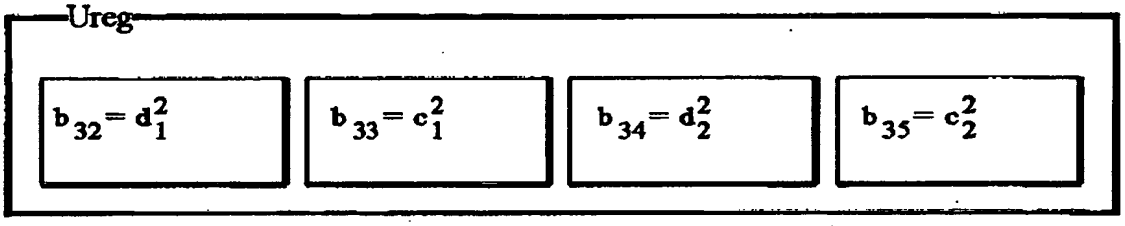
【図２９】



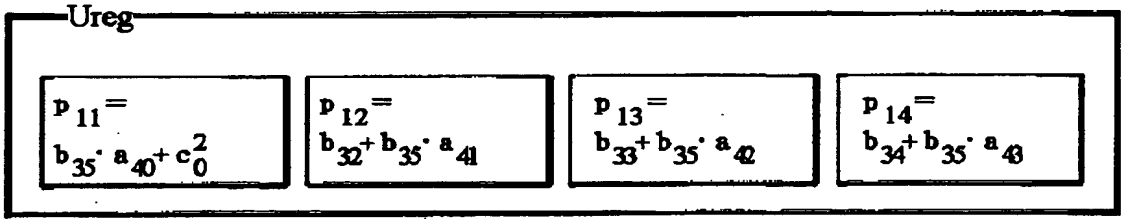
【図３０】



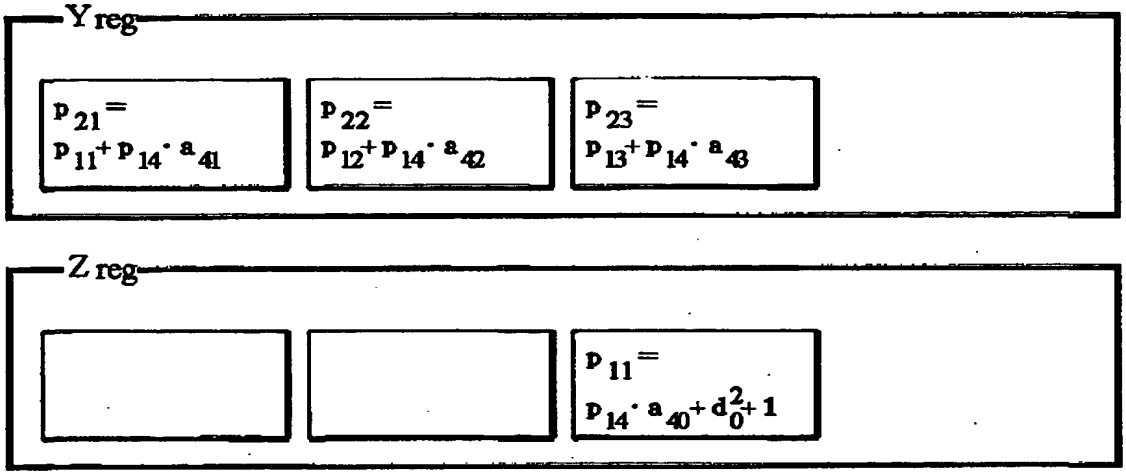
【図３１】



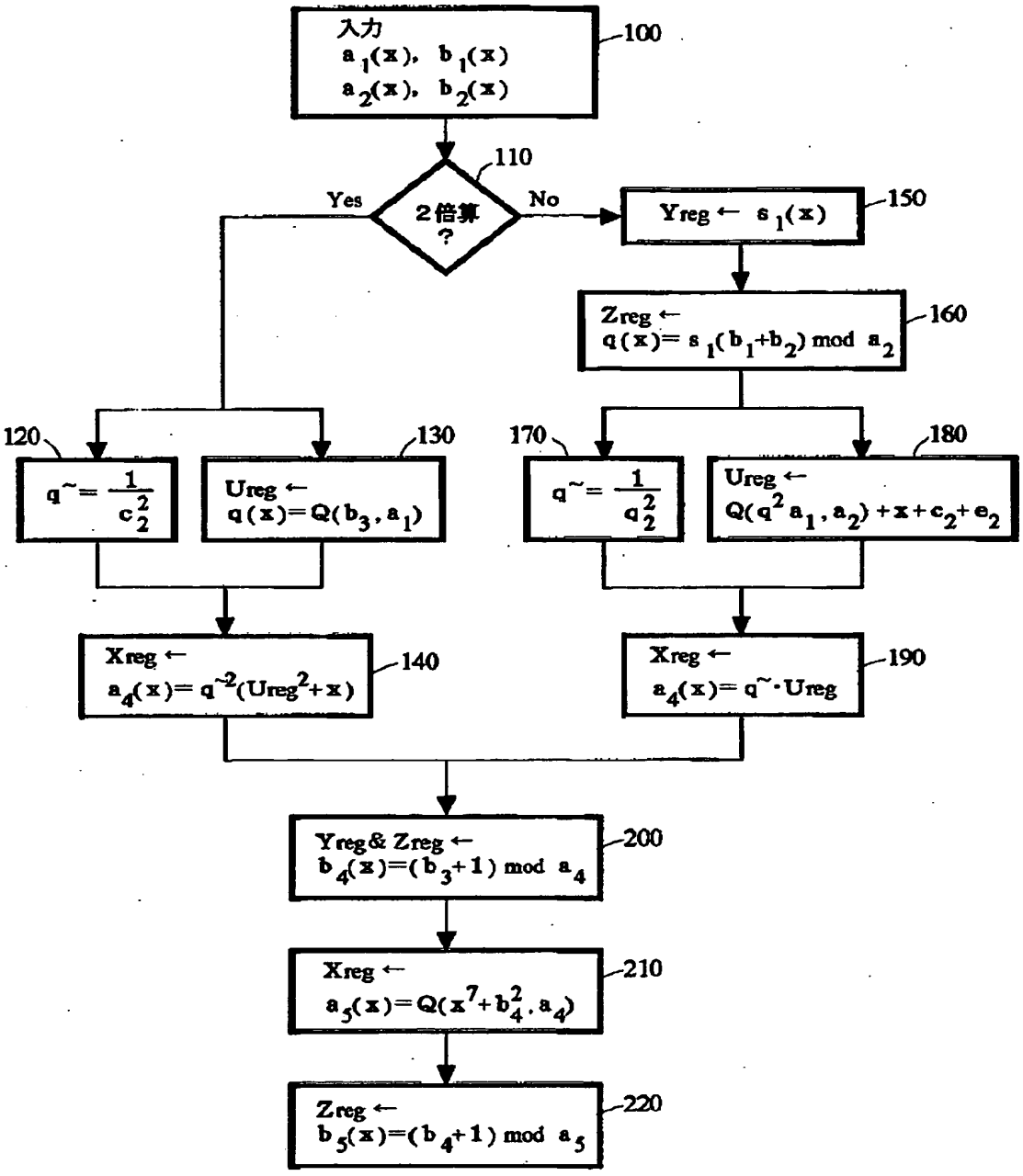
【図３２】



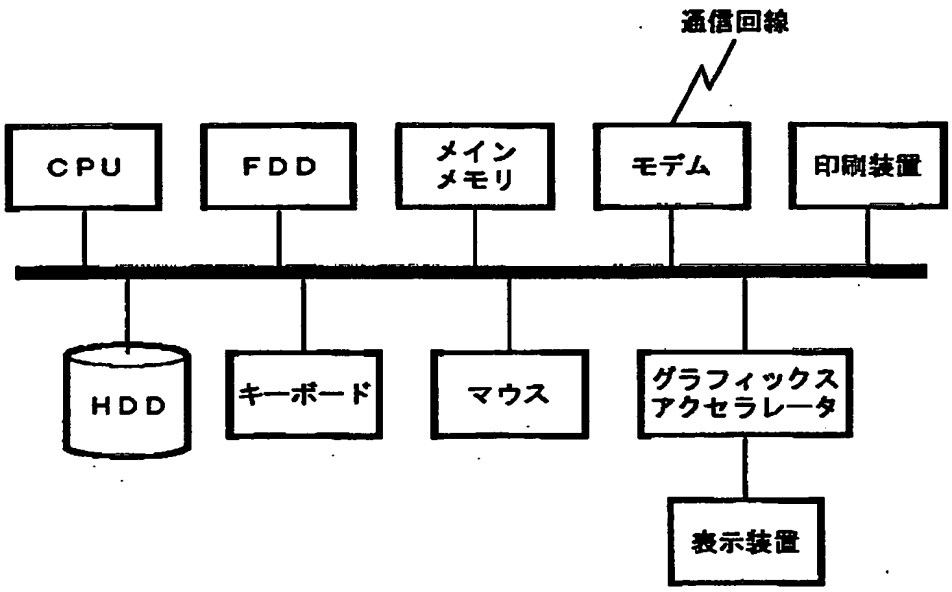
【図３３】


Y reg
$p_{21}=p_{11}+p_{14}\cdot a_{41}$
$p_{22}=p_{12}+p_{14}\cdot a_{42}$
$p_{23}=p_{13}+p_{14}\cdot a_{43}$
Z reg
$p_{11}=p_{14}\cdot a_{40}+d_0^2+1$

【図３４】


入力
$a_1(x)$, $b_1(x)$
$a_2(x)$, $b_2(x)$
100
110
2倍算 ?
Yes
No
Yreg ← $s_1(x)$
150
Zreg ←
$q(x) = s_1(b_1+b_2) \bmod a_2$
160
120
$q^{\sim} = \frac{1}{c_2^2}$
130
Ureg ←
$q(x) = Q(b_3, a_1)$
170
$q^{\sim} = \frac{1}{q_2^2}$
180
Ureg ←
$Q(q^2 a_1, a_2) + x + c_2 + e_2$
Xreg ←
$a_4(x) = q^{\sim 2}(Ureg^2 + x)$
140
Xreg ←
$a_4(x) = q^{\sim} \cdot Ureg$
190
Yreg& Zreg ←
$b_4(x) = (b_3+1) \bmod a_4$
200
Xreg ←
$a_5(x) = Q(x^7 + b_4^2, a_4)$
210
Zreg ←
$b_5(x) = (b_4+1) \bmod a_5$
220

【図３５】


通信回線
CPU
FDD
メイン
メモリ
モデム
印刷装置
HDD
キーボード
マウス
グラフィックス
アクセラレータ
表示装置

【書類名】　要約書

【要約】

【課題】

ヤコビ多様体における群演算を少ない演算量にて実現すること。

【解決手段】

ＧＦ（2^n）上で定義される超楕円曲線$y^2+y=f(x)$のヤコビ多様体の因子D_1=g.c.d.（$a_1(x)$，$y-b_1(x)$）及びD_2=g.c.d.（$a_2(x)$，$y-b_2(x)$）に対し群演算を実施する装置は、$a_1(x)$、$a_2(x)$、$b_1(x)$及び$b_2(x)$を格納する手段と、GCD（$a_1(x)$，$a_2(x)$）＝１（GCDは最大公約多項式）である場合における$s_1(x)a_1(x)+s_2(x)a_2(x)=1$となる$s_1(x)$を用いて、ｑ（ｘ）＝｛$s_1$（$b_1(x)+b_2(x)$）｝ｍｏｄ　$a_2(x)$を計算する手段と有する。このように新たな関数ｑ（ｘ）を設けることにより、計算量が減少し、且つハードウエア量も少なくて済む。なお、$D_1=D_2$の場合には、$a_1(x)$及び$b_1(x)$を格納する手段と、ｑ（ｘ）＝Ｑ（$b_1^2(x)+f(x)$ mod $a_1^2(x)$，a_1）（Ｑ（Ａ，Ｂ）はＡをＢで割ったときの商）を計算する手段を設ける。

【選択図】　図１

認定・付加情報

特許出願の番号	平成１１年　特許願　第００７３８４号
受付番号	５９９０００２９７５９
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１１年　１月２４日

＜認定情報・付加情報＞

【提出日】　平成11年 1月14日

次頁無

出証特平１１－３０２７５２０

出　願　人　履　歴　情　報

識別番号　　〔３９０００９５３１〕

１．変更年月日　　１９９０年１０月２４日
［変更理由］　新規登録
住　所　　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）
氏　名　　インターナショナル・ビジネス・マシーンズ・コーポレイション